

PLANIFICACIÓN DE CAMINOS  
PARA ROBOTS INDUSTRIALES  
BASADA EN EL ALGORITMO DE LAS  
HORMIGAS



UNIVERSIDAD CARLOS III DE MADRID

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

Autor: Adrián Rodríguez Sánchez

Tutor: Fares Jawad Abu-Dakka

# **Planificación de Caminos para Robots Industriales Basada en el Algoritmo de las Hormigas**

Autor: Adrián Rodríguez Sánchez

Tutor: Fares Jawad Abu-Dakka

## **MIEMBROS DEL TRIBUNAL**

**PRESIDENTE:** Luis Santiago Garrido Bullón

**SECRETARIO:** Jorge Sánchez Ruiz

**VOCAL:** Maria Fernanda Mendoza Petit

Acto de defensa y lectura del Trabajo de Fin de Grado el día 7 de Julio de 2016, en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid.



# Agradecimientos

Antes de comenzar, me gustaría agradecer a todas las personas que me han ayudado a llegar hasta aquí. Este trabajo de fin de grado no sólo es el resultado de mi esfuerzo, sino de toda la gente que me rodea y me ha apoyado a seguir adelante.

En primer lugar, quisiera agradecerle a mi tutor Fares Abu-Dakka todo el esfuerzo y dedicación que ha puesto para ayudarme con este proyecto. Siempre ha estado disponible cuando lo he necesitado y me ha orientado pacientemente en todas las dificultades que me han surgido.

A mis padres, por enseñarme a no rendirme y buscar siempre lo mejor de mí. A mi hermana Irene, por ser un modelo a seguir. A mi hermano Sergio, que siempre ha estado apoyándome en todos los buenos y malos momentos.

# Índice de contenidos

<b>Índice de figuras .....</b>	<b>8</b>
<b>Índice de tablas .....</b>	<b>10</b>
<b>Glosario de acrónimos.....</b>	<b>12</b>
<b>1. INTRODUCCIÓN .....</b>	<b>15</b>
1.1. Perspectiva clásica de la planificación de caminos.....	16
1.2. Algoritmo de las hormigas .....	19
1.3. Objetivos del Proyecto .....	22
1.4. Estructura de la memoria del proyecto.....	23
<b>2. MODELADO DEL ROBOT Y DEL ESPACIO DE TRABAJO .....</b>	<b>25</b>
2.1. Modelado del sistema del robot .....	26
2.2. Problema cinemático .....	27
2.2.1. Introducción a la cinemática .....	28
2.2.2. Cinemática directa .....	29
2.2.3. Aplicación de la cinemática directa en Puma un 560 .....	30
2.2.4. Cinemática inversa.....	31
2.2.5. Aplicación de la cinemática inversa en un Puma 560 .....	32
2.3. Modelado del entorno.....	35
2.3.1. Modelado del espacio de trabajo .....	35
2.3.2. Modelado de obstáculos.....	37
2.4. Cálculo de distancias mínimas de colisión.....	38
2.4.1. Distancias mínimas con esferas .....	38
2.4.2. Distancias mínimas con cilindros .....	40

2.4.3. Distancias mínimas con planos.....	43
<b>3. EL ACO EN LA PLANIFICACIÓN DE CAMINOS .....</b>	<b>47</b>
3.1. Perspectiva general del ACO .....	47
3.1.1. Introducción de la optimización .....	49
3.1.2. Explicación del algoritmo.....	50
3.1.3. Ejemplo práctico .....	51
3.1.4. Variaciones .....	53
3.1.4.1. Sistema de hormigas elitista (EAS) .....	53
3.1.4.2. Sistema de hormigas MAX-MIN (MMAS).....	54
3.1.4.3. Sistema de hormigas basado en rankings ( $AS_{rank}$ ).....	54
3.1.4.4. Sistema colonias de hormigas (ACS).....	55
3.2. Obtención de caminos mediante ACO .....	56
<b>4. ESTIMACIÓN ESTADÍSTICA DE LOS PARÁMETROS DEL ACO.....</b>	<b>60</b>
4.1. Análisis de varianza (ANOVA) .....	61
4.2. Entorno desarrollado para el estudio.....	61
4.3. Resultados obtenidos.....	62
4.3.1. Varianza del número de hormigas .....	62
4.3.2. Varianza del número de iteraciones.....	65
4.3.3. Estudios de $\alpha$ , $\beta$ y $\rho$ .....	68
4.3.3.1. Varianza de $\alpha$ .....	68
4.3.3.2. Varianza de $\beta$ .....	72
4.3.3.3. Varianza de $\rho$ .....	75
4.4. Discusión de los resultados .....	77
<b>5. EXPERIMENTOS DE PLANIFICACIÓN DE CAMINOS.....</b>	<b>79</b>

5.1. Ejemplo 1. Obstáculos sencillos.....	79
5.2. Ejemplo 2. Aplicación industrial.....	83
5.3. Ejemplo 3. Obstrucción de rutas .....	85
<b>6. CONCLUSIÓN .....</b>	<b>88</b>
<b>7. REFERENCIAS.....</b>	<b>91</b>
<b>Apéndice A. Presupuesto .....</b>	<b>95</b>

# Índice de figuras

Figura 1. Modelado alámbrico del robot con espacio de trabajo y puntos significativos y de interés. ....	27
Figura 2. Notación modificada de Denavit y Hartenberg.....	29
Figura 3. Articulaciones y ejes del Puma 560. ....	31
Figura 4. Ejemplo espacio de trabajo. Un prisma, donde los puntos extremos de la diagonal principal son el punto de inicio y el de destino. Desde un nodo se puede acceder como máximo a otros siete mediante sus arcos.....	36
Figura 5: Tipos de obstáculos y datos que los definen. ....	38
Figura 6. Distancias mínimas con esferas .....	39
Figura 7. Diagrama de flujo colisiones con esferas.....	40
Figura 8. Distancias mínimas con cilindros.....	41
Figura 9. Diagrama de flujo colisiones con cilindros.....	43
Figura 10. Distancias mínimas con planos. ....	44
Figura 11. Diagrama de flujo colisiones con planos. ....	46
Figura 12. Experimento “Self-organized Shortcuts in the Argentine Ant” .....	48
Figura 13. Ejemplo de grafo.....	49
Figura 14. Grafo del TSP.....	51
Figura 15. TSP realizado por la primera hormiga .....	52
Figura 16. TSP realizado por la segunda hormiga.....	52
Figura 17. Pseudocódigo de la generación parcial del espacio operativo. ....	57
Figura 18. Pseudocódigo de ACO. ....	59
Figura 19. Distancia óptima en función del número de hormigas.....	63
Figura 20. Tiempo de ejecución en función del número de hormigas. ....	63
Figura 21. Distancia óptima en función del número iteraciones. ....	66



Figura 22. Tiempo de ejecución en función del número de iteraciones.....	66
Figura 23. Distancia óptima en función de $\alpha$ .....	68
Figura 24. Tiempo de ejecución en función de $\alpha$ .....	69
Figura 25. Distancia óptima en función de $\beta$ .....	72
Figura 26. Tiempo de ejecución en función de $\beta$ .....	72
Figura 27. Distancia óptima en función de $\rho$ .....	75
Figura 28. Tiempo de ejecución en función de $\rho$ .....	75
Figura 29. Entorno con esferas.....	81
Figura 30. Entorno con cilindros.....	81
Figura 31. Entorno con planos.....	82
Figura 32. Movimiento del robot por la ruta establecida como óptima.....	84
Figura 33. Resultados del ejemplo de obstrucción de rutas, mostrando la distancia de cada una de las rutas en función de los obstáculos definidos.....	86
Figura 34. Evolución de las rutas en el entorno. Se muestran todos los obstáculos con 5 rutas distintas para facilitar su visualización.....	87

# Índice de tablas

Tabla 1. Parámetros para el ejemplo de TSP.....	51
Tabla 2. Parámetros fijados para el estudio de la influencia del número de hormigas. ....	62
Tabla 3. Distancia óptima en función del número hormigas.....	64
Tabla 4. Tiempo de ejecución en función del número hormigas.....	65
Tabla 5. Parámetros fijados para el estudio de la influencia del número de iteraciones.....	65
Tabla 6. Distancia óptima en función del número iteraciones. ....	67
Tabla 7. Tiempo de ejecución en función del número iteraciones. ....	67
Tabla 8. Parámetros fijados para el estudio de la influencia de $\alpha$ , $\beta$ o $\rho$ . ....	68
Tabla 9. Distancia óptima en función de $\alpha$ . ....	70
Tabla 10. Tiempo de ejecución en función de $\alpha$ . ....	71
Tabla 11. Distancia óptima en función de $\beta$ . ....	73
Tabla 12. Tiempo de ejecución en función de $\beta$ . ....	74
Tabla 13. Distancia óptima en función de $\rho$ . ....	76
Tabla 14. Tiempo de ejecución en función de $\rho$ . ....	76
Tabla 15. Valores óptimos de estadísticos.....	78
Tabla 16. Parámetros del ACO para el ejemplo 1. ....	80
Tabla 17. Configuración inicial y final del robot. ....	80
Tabla 18. Ubicación de los obstáculos. ....	80
Tabla 19. Resultados ejemplo de obstáculos sencillos comparado con el artículo de (Abu-Dakka, et al. 2012), donde se usan algoritmos genéticos.....	82
Tabla 20. Parámetros del ACO para el ejemplo 2. ....	83
Tabla 21. Ubicación de los obstáculos. ....	83
Tabla 22. Configuración inicial y final del robot. ....	84

Tabla 23. Resultados ejemplo aplicación industrial, comparado con el artículo de (Abu-Dakka, et al. 2012), donde se usan algoritmos genéticos. ....	84
Tabla 24. Configuración inicial y final del robot. ....	85
Tabla 25. Ubicación de los obstáculos. ....	85
Tabla 26 Parámetros del ACO para el ejemplo 3. ....	86
Tabla 27. Horas estimadas de trabajo en el proyecto. ....	96
Tabla 28. Coste del Proyecto. ....	96

# Glosario de acrónimos

ACO	Ant Colony Optimization	Algoritmo de las Hormigas
ACS	Ant Colony System	Sistema de Colonia de Hormigas
ANOVA	Analysis Of Variance	Análisis de Varianza
APF	Artificial Potential Field	Campo Potencial Artificial
ASrank	Ant System Rank	Sistema de Hormigas basado en Rankings
EAS	Elitist Ant System	Sistema Elitista de Hormigas
GDL	Grados De Libertad	
m-DH	Modified Denavit-Hartenberg	Denavit-Hartenberg Modificado
MMAS	Max Min Ant System	Sistema Max-Min de Hormigas
PRM	Probabilistic Roadmap Method	Método Probabilístico de Mapa de carreteras
RPP	Randomized Path Planner	Planeador Aleatorio de Caminos
RRT	Rapidly exploring Random Tree	Árboles Aleatorios de Exploración Rápida
TFG	Trabajo Fin de Grado	
TSP	Traveling Salesman Problem	Problema del Viajante

# Resumen

Este trabajo Fin de Grado presenta un algoritmo de optimización de caminos libres de obstáculos para robots industriales. Su metodología está basada en el algoritmo de las hormigas - Ant Colony Optimization (ACO)-, aplicado a un espacio de trabajo discretizado en tres dimensiones, con un sistema de coordenadas cartesianas, formado por el punto de partida, por el punto de destino y por los obstáculos.

El propósito de este algoritmo es obtener la ruta más corta entre dos configuraciones (inicial y final), evitando obstáculos y colisiones entre las configuraciones del robot y los elementos del entorno. Adicionalmente, se detallan una serie de análisis de varianza ANOVA para determinar la influencia directa que tienen los parámetros del ACO ( $\alpha$ ,  $\beta$ ,  $\rho$ , número de hormigas y número de iteraciones) sobre el camino óptimo encontrado y el tiempo de ejecución del algoritmo.

La simulación de los ejemplos se ha desarrollado, en distintos entornos, para un robot Puma 560, y haciendo frente a diferentes obstáculos que dificultan algunas de las rutas. Además, se ha comparado el algoritmo propuesto con otros trabajos publicados, mostrando una mejora significativa en la optimización de las rutas.

## *Palabras Clave:*

Robots industriales, Algoritmo de las Hormigas, Evitación de Obstáculos, Planificación de caminos.

# English Summary

## **Path Planning Optimization for Industrial Robots Using Ant Colony Optimization.**

### **Abstract**

The present work describes the development of an algorithm for path planning optimization, designed for industrial robots, which allows the avoidance of obstacles. The detailed methodology is based on the Ant Colony Optimization algorithm, applied to a 3D discrete workspace (defined in Cartesian coordinates), which includes the starting and destination configurations as well as several obstacles.

The goal of this algorithm is to obtain the shortest path between two configurations (starting and final), avoiding obstacles and collisions between different configurations. In addition, we define several analyses of variance ANOVA to determine the direct effect that ACO's parameters ( $\alpha$ ,  $\beta$ ,  $\rho$ , number of ants and number of iterations) have on the optimal path obtainment and the required computational time.

The simulation described has been optimized for a Puma 560, and uses several environments and contains multiple obstacles that interfere with various paths. Furthermore, the present work represents a significant improvement compared to previously reported path planning optimization algorithms.

### *Keywords:*

Industrial robots, Ant Colony Optimization, Obstacles avoidance, Path planning.

# CAPÍTULO 1

## Introducción

En el ámbito de la robótica, la necesidad de perfeccionar las rutas de desplazamiento (planificación de caminos o bien conocido como *path planning*) ha sido siempre un objetivo destacado en numerosas investigaciones y una demanda continua de los operadores y usuarios. Los nuevos desarrollos que se llevan a cabo buscan dotar al robot de una gran autonomía, consiguiendo así que pueda desplazarse por su espacio operativo sin necesidad de que intervenga el programador, dando soluciones, de forma autónoma, a situaciones imprevistas o estableciendo tareas personalizadas bajo demanda del operador.

Obviamente, la técnica de planificación de caminos, busca una ruta libre de colisiones -desde un inicio hasta un destino- en un entorno con obstáculos, intentando optimizar algún criterio clave, como acortar las distancias recorridas, o emplear el menor tiempo posible en el itinerario o reducir los costes operativos, entre otros.

Existen multitud de estudios enfocados a la planificación de caminos. Según (Hwang and Ahuja 1992) se pueden clasificar en dos ramas: globales y locales. Para los métodos globales es necesario un conocimiento completo del entorno y que éste sea estático, de modo

que el resultado es una ruta optimizada antes de que el robot empiece el recorrido. Por el contrario, en los métodos locales, el algoritmo optimiza nuevos caminos a medida que el robot se mueve por uno ya definido, es decir, permite el estudio en entornos dinámicos.

Algunos ejemplos de algoritmos orientados a la planificación de caminos son: el algoritmo A\* (Warren 1993), método Probabilístico de carreteras (Kavraki et al. 1996) el cual es conocido por sus términos en inglés como Probabilistic Roadmap Method (PRM), algoritmos genéticos (Abu-Dakka et al. 2013, Abu-Dakka et al. 2012), redes neuronales (Dahl 1987), el método de campos de potencial artificial conocido como Artificial Potencial Field (APF) (Khatib 1986, Li et al. 2004), y el algoritmo de las hormigas o Ant Colony Optimization (ACO) (Dorigo et al. 2008). Todos ellos presentan alguna ventaja e inconveniente que hace más atractiva su elección dependiendo del problema al que se va a hacer frente. Por ejemplo, el APF tiene como desventajas que cuando el robot se encuentra muy alejado del destino, la fuerza de atracción, comparada con la de repulsión ejercida por los obstáculos, es muy elevada. Esto provoca que se mueva muy cerca de ellos. Además, los obstáculos cerca de los destinos pueden hacerlos inalcanzables. Si la fuerza de atracción y repulsión se equilibran, provocarán que el robot tenga un campo potencial nulo, lo que le creará un bloqueo en un mínimo local.

Un inconveniente de las redes neuronales es que no pueden reentrenarse con datos posteriores en redes existentes. Por otro lado, la gestión de series de tiempo que necesitan las redes neuronales es una complejidad añadida cuando se usan estos algoritmos.

## **1.1. Perspectiva clásica de la planificación de caminos**

Como se ha mencionado la subsección anterior, para resolver problemas de planificación de caminos contamos con una gran variedad de métodos. No obstante, en esta sección se hablará de algunos de los métodos clásicos/básicos empleados en la historia de



planificación de caminos. Estos se pueden dividir en dos tipos: aproximación a “mapas de carreteras” (Roadmap Approach) y métodos “probabilísticos de planificación de caminos”.

En las aproximaciones a “mapa de carreteras”, la idea básica consiste en capturar la conectividad del espacio libre del robot  $C_f$ , mediante un grafo o una red de caminos. Formando así un conjunto de nodos en esta red, que estarán conectados si existe un camino libre entre ellos. Una vez se construye este mapa, se buscan caminos entre un inicio y un destino. Varios métodos que utilizan esta aproximación se describen a continuación.

El primero de ellos es el método gráfico de visibilidad, que es uno de los más antiguos para el estudio de caminos. Propuesto por (Hart et al. 1968), el mapa se genera a partir de obstáculos poligonales, cuyos vértices corresponden con los nodos por los que se pueden formar caminos. Se encuentran conectados si se hallan en la misma arista del polígono o, si por el espacio, se pueden unir sin ningún corte provocado por otro obstáculo. Una vez se define todo el mapa de carreteras se pueden emplear métodos como el algoritmo A\* para obtener el camino más corto.

El llamado diagrama Voronoi, básicamente consiste en posicionar sobre una superficie, un conjunto de puntos (Klee 1980). Estos tienen una región asignada de forma que lo que se encuentra en esta ubicación está más cerca de este punto que de cualquier otro. Por ejemplo un mosaico de pentágonos, donde el punto se posicionaría en el centro de cada pentágono y su región sería el área contenida en ellos. Su resolución se puede implementar con algoritmos de tipo “Fast marching” (Osher and Sethian 1988), donde su metodología consiste en la expansión o contracción de los bordes de una región respecto a una función de velocidad.

“Freeway”, (Brooks 1983), a diferencia de los anteriores, define los “caminos libres” por un campo potencial artificial que se produce por una repulsión entre las superficies de los obstáculos.

Con este procedimiento de (Khatib 1986) se establecen en todo el espacio operativo dos tipos de fuerzas: las de atracción y las de repulsión. Las de atracción son ejercidas por los

puntos de destino, y las de repulsión son generadas por los obstáculos. En un nodo aparece un gradiente que induce una dirección hacia el destino, evitando los obstáculos. La gran ventaja que tiene esta metodología es que se puede aplicar en la planificación de caminos locales (entornos dinámicos).

El segundo de los métodos clásicos son los llamados “probabilísticos de planificación de caminos”. Presentan la solución a los elevados tiempos de computación que van ligados a la representación explícita de grandes espacios de trabajos. Se basan en comprobar si una configuración específica es accesible, sin necesidad de representarla. Los procedimientos más utilizados en los últimos años han sido: el método probabilístico de mapa de carreteras (PRM), los árboles aleatorios de exploración rápida (RRT) y el planeador aleatorio de caminos (RPP).

El método probabilístico de mapa de carreteras (PRM), fue desarrollado por (Barraquand and Latombe 1991). En una primera fase se generan aleatoriamente  $n$  configuraciones para el robot y se verifican sus conexiones. De este modo, se genera un espacio definido por nodos y arcos, calculados a lo largo de todo el espacio de manera aleatoria. Posteriormente, se inicia la fase de consulta, donde se intenta conectar la configuración inicial y final. La gran ventaja que posee este método es que ambas fases (generación y consulta) pueden realizarse simultáneamente. La generación puede interrumpirse cuando se encuentra una solución, consiguiéndose así tiempos de computación muy pequeños.

El sistema de árboles aleatorios de exploración rápida (RRT), desarrollado por (LaValle 1998) consiste en generar, de forma aleatoria, una estructura basada en un “árbol” desde un nodo inicial. Se generan así, desde el punto inicial, una serie de ramas en direcciones radiales. Desde ellos se ramifican otros que podrán a su vez crear nuevos. La gran desventaja que posee esta metodología es que, normalmente, su uso en solitario suele ser insuficiente para resolver un problema de planificación, por lo que conviene acoplarlo con alguna otra optimización.

Finalmente, se describe el método planeador aleatorio de caminos (RPP), desarrollado por (Barraquand and Latombe 1991), y es conocido como la primera generación de los métodos probabilísticos de planificación de caminos. Al igual que el PRM, presenta dos fases, una de generación y otra de consulta. Durante la generación, se crean aleatoriamente caminos entre las dos configuraciones –inicial y final- y a su vez se van conectando todos los nodos, si estos disponen de un arco libre de colisión. Durante la fase de consulta los nodos inicial y final se conectan a dos nodos aleatorios del mapa. Estos a su vez, eligen otros dos nodos disponibles, repitiéndose esta secuencia hasta que finalmente se forme un camino.

## 1.2. Algoritmo de las hormigas

El algoritmo empleado en este Proyecto Fin de Grado se basa en el ACO propuesto por (Dorigo 1992), aunque presenta el gran inconveniente de no poder aplicarse con facilidad a entornos dinámicos (planificación de caminos local). Es un algoritmo heurístico que ha dado magníficos resultados en lo que respecta a optimización de caminos (Tewolde and Sheng 2008).

El ACO, como su propio nombre sugiere, está basado en el comportamiento que siguen las hormigas en sus rutas desde el hormiguero hasta los emplazamientos donde se ubican alimentos. Sus principios básicos son los siguientes:

1. Siguen rutas aleatorias hasta encontrar comida.
2. De regreso a la colonia, dejan un rastro de feromona.
3. Esta traza puede ser detectada por otra hormiga, que la reforzará si encuentra comida.
4. La feromona se va evaporando.

El ACO se ha aplicado en numerosas ocasiones para resolver problemas de planificación de caminos, como en robots humanoides (Zhang and Zhong 2014), logística de fabricación (Guo et al. 2014), robots móviles (Garro et al. 2007), incluso evitando la colisión múltiple entre varios robots móviles (Zhou et al. 2014).

Algunos autores combinan el ACO junto con otra metodología de optimización, por ejemplo, en (Mei et al. 2006) se combina el ACO con el APF. Como resultado se presenta un algoritmo con respuesta a entornos dinámicos. Además, realiza una modificación en lo que respecta a la distribución de feromona inicial: estos niveles no se distribuyen por igual en todo el espacio, sino que, cuanto más cerca se encuentren los puntos de un obstáculo, menos concentración dispondrán en un inicio. Así se consigue que el algoritmo converja en menos tiempo en una solución. Por otro lado, si el camino óptimo tiene que pasar cerca de algún obstáculo, esta proximidad dificultará su elección, provocando que no se llegue a alcanzar la mejor ruta o incluso que se tenga que potenciar la exploración añadiendo más hormigas o incrementando el ratio de evaporación, para intentar así mejorar la probabilidad de elección de estos puntos en un inicio.

En (Garro, et al. 2007), con el fin de optimizar rutas para robots móviles, se combina el ACO con un algoritmo genético. Algunos cambios que se incluyen son la utilización de una nueva regla de transición entre nodos y la creación de nuevas hormigas (con cierta frecuencia), empezando todas ellas en el nodo inicial. Si alguna de estas supera una cierta distancia en lo que respecta a su ruta, se elimina. Con esto lo que se consigue es que no se tengan en cuenta ciertas rutas que exceden un valor predeterminado. Toda esta metodología se puede comparar con la variación del Sistema de hormigas basado en rankings, ya que, estableciendo un tamaño de ranking en el que numerosas hormigas no aporten feromona, se consigue eliminar el impacto que generan las peores, o sea, las que están por debajo del umbral preestablecido, sin necesidad de crear nuevos individuos. Esto evitaría que, la población bajase de un determinado número en el cual no puedan converger en una solución eficiente, por imposibilitar la exploración al no contar con demasiadas hormigas. Es por eso que, en este Proyecto Fin de Grado se incorpora un estudio estadístico que muestra el impacto de los parámetros del ACO.

Centrándonos en el tema de manipuladores robóticos en espacios de tres dimensiones, tenemos a (Zhou et al. 2016), definiendo un problema con obstáculos y proponiendo algunas modificaciones al ACO para evitar caer en mínimos locales. Se formula una ecuación de transición -como en el del Sistema Colonia de Hormigas-, con un nuevo parámetro aleatorio que puede modificarse con el objetivo de favorecer la exploración. Los resultados de las

simulaciones demuestran que la modificación frente al ACO estándar mejora sensiblemente el rendimiento. No obstante, los estudios realizados no contemplan la simulación de obstáculos complejos, que planteen un entorno más delicado y por otro lado, la incorporación del parámetro aleatorio puede sustituirse en gran medida por un incremento en el número de hormigas para facilitar la exploración.

Por otro lado, siguiendo con los manipuladores en 3D (Zhao and Yong 2015) utiliza la actualización de control borrosa (Fuzzy Control Updating) para renovar los valores de feromona de la colonia. Cabe destacar la modificación que aplica al colocar los niveles de feromona en los puntos, y no en los arcos. Esto, en general, supone una mejora con la que se consigue que el algoritmo converja en menos tiempo. No obstante, aunque el algoritmo encuentre siempre la mejor solución, si se requiere continuidad en el uso, y se ha encontrado más de un camino óptimo (misma longitud), puede llegar a producir conflictos, posibilitando que se generen multitud de rutas que combinando estas óptimas, den como resultado una que no lo es.

Numerosos estudios han demostrado la viabilidad del ACO para la resolución de distintos problemas combinatorios, ya sean de tipo dinámico, multi-objetivo, problemas del viajante u otros (Mohammed 2013, Sariff and Buniyamin 2010). La mayoría se plantean en un sistema plano, donde la caracterización del espacio a analizar resulta menos compleja.

El algoritmo que se define en este proyecto incorpora la posibilidad de resolver problemas en tres dimensiones, aunque no sea una innovación ya que algunos otros tienen esa propiedad (Zhao and Yong 2015). Sus tamaños suelen ser entornos pequeños, pero en nuestro algoritmo planteamos el modelado de forma que se puedan generar espacios relativamente grandes (9261 nodos en los estudios estadísticos), obviamente, incrementando el tiempo de ejecución. También incluye soluciones para sortear los obstáculos definidos en el entorno y que se puedan encontrar en el recorrido (Zhou, et al. 2016). Para facilitar el posicionamiento de los estorbos, el planteamiento del espacio de trabajo se realiza en un sistema de coordenadas cartesianas.

Desde que Dorigo planteó el algoritmo, se han desarrollado numerosas variaciones del ACO que, de una forma u otra, pueden optimizar los resultados obtenidos en la ejecución del algoritmo. Algunos ejemplos son:

- Sistema Colonia de Hormigas (Ant Colony System, ACS) (Chakraborty 2013): incorpora una nueva regla de transición y restricciones en la actualización de la feromona.
- Sistema de Hormigas Máximo – Mínimo (Max – Min Ant System, MMAS) (Stützle and Hoos 2000): el valor memorístico (feromona) está limitado por un máximo y un mínimo.
- Sistema de Hormigas Elitistas (Elitist Ant System, EAS)(Negulescu et al. 2008): su principal característica es que da un mayor peso al mejor camino encontrado hasta el momento.
- Sistema de Hormigas Basado en Rankings (Rank-Based Ant System, ASrank) (Bullnheimer et al. 1997): establece un ranking de hormigas basándose en la longitud de los caminos que recorren por lo que, cuando se encuentran en una posición más elevada del ranking, más valor memorístico depositarán en los arcos.

La reciente proliferación de robots manipuladores en multitud de entornos, ha potenciado el estudio de algoritmos planificadores de rutas en los últimos años. Aunque el ACO empezó a desarrollarse en problemas del tipo viajante (Travelling Salesman Problem, TSP) (Dorigo et al. 1997), su evolución se ha ramificado hacia numerosos campos.

## 1.3. Objetivos del proyecto

El estudio planteado en este proyecto consiste en, mediante la planificación de caminos, poder determinar unas rutas óptimas entre una configuración inicial y final de un manipulador robótico, todo ello en tres dimensiones y haciendo uso del ACO. A su vez, se propone determinar una relación entre los parámetros del ACO y su efectividad a la hora de definir una solución.

En línea con lo anterior, se hace uso de la Colonia Simple de Hormigas (el ACO sin ninguna de sus variaciones) con alguna pequeña modificación para simplificar su adaptación a un entorno de tres dimensiones.

Se han realizado dos experimentos para comparar este algoritmo con el algoritmo genético usado en (Abu-Dakka, et al. 2012), habiéndose definido para ambos las mismas condiciones en el entorno y parámetros de generación de espacio (pasos del robot). En otro experimento realizado, se demuestra la viabilidad del algoritmo para hacer frente a nuevos obstáculos en cada una de sus rutas calculadas. Por último, se detallan una serie de estudios estadísticos de varianza ANOVA, donde se puede evaluar la influencia de cada parámetro del ACO.

## **1.4. Estructura de la memoria del proyecto**

A modo de resumen, a continuación se detalla lo que contiene cada capítulo de esta memoria.

En el capítulo 2, empezando por resolver el problema de la cadena cinemática se formulan todas las ecuaciones cinemáticas que se utilizan para el robot Puma 560, como un ejemplo de su aplicación. A su vez, se detalla el cálculo de distancias mínimas con los obstáculos y el robot con el fin de evitar colisiones.

En el capítulo 3, se explica de dónde surge el ACO, cómo funciona y se muestra un pequeño ejemplo para la resolución de un problema del tipo viajante. A su vez, se detalla cómo se ha aplicado el algoritmo para optimizar las rutas con manipuladores.

El capítulo 4, tiene los estudios estadísticos que se han realizado con el fin de intentar optimizar los parámetros en el algoritmo.

El capítulo 5, contiene los tres experimentos que se han desarrollado utilizando el algoritmo con el Puma 560, obstáculos sencillos, entorno industrial y flexibilidad para encontrar caminos.

Por último, en el capítulo 6 se explican las conclusiones a las que se ha llegado en el proyecto. Así como algunas ideas para el futuro y cambios que pueden mejorar el algoritmo.



# Capítulo 2

## Modelado del robot y del espacio de trabajo

Para que el algoritmo pueda analizar el entorno y buscar una solución óptima, es necesario extrapolar un modelo en el que se puedan cuantificar una serie de características que influyan en las decisiones tomadas por el algoritmo. De esta forma, se establece una ramificación para encontrar el camino óptimo y libre de obstáculos entre dos configuraciones dadas: una inicial y otra final.

En este capítulo se detalla la formulación del problema cinemático del robot -un Puma 560-, así como el proceso para esquivar los obstáculos del entorno, ya sea por colisión de cualquiera de los brazos del robot o porque el punto sea inaccesible.

## 2.1. Modelado del sistema del robot

El modelado del robot consiste en un modelo alámbrico, formado por una serie de articulaciones que mueven las uniones rígidas del robot y se define en función de coordenadas generalizadas.

Para facilitar el cálculo de colisiones, el espacio de trabajo y los obstáculos se definen en coordenadas cartesianas, aunque las configuraciones del robot se expresan en coordenadas articuladas  $C^i(\mathbf{q})$ ;  $\mathbf{q} = [q_1, q_2, \dots, q_j, \dots, q_n]^T$ , donde  $n$  representa el número de grados de libertad del robot.

En el modelado del robot, una configuración  $j$  se define mediante una serie de puntos significativos  $\boldsymbol{\varphi}_k^j$  y otros de interés  $\boldsymbol{\varepsilon}_k^j$ , donde  $k$  represente el número de dichos puntos, expresados en coordenadas cartesianas y como función de coordenadas generalizadas. Haciendo uso de estos puntos, las coordenadas articulares del robot se puedan expresar en coordenadas cartesianas. Esos puntos, también facilitan el cálculo de colisiones del robot con los posibles obstáculos definidos en el espacio operacional.

El posicionamiento y declaración de estos puntos significativos y de interés no es casual porque depende del número de GDL del robot y de la geometría de éste. El resultado debe ser tal que, con el mínimo número de puntos, se pueda definir con precisión una configuración del sistema robótico.

En la Figura 1 se muestra un ejemplo con 4 puntos significativos  $\boldsymbol{\varphi}_k^j = (\boldsymbol{\varphi}_1^j, \boldsymbol{\varphi}_2^j, \boldsymbol{\varphi}_3^j, \boldsymbol{\varphi}_4^j)$  y 4 puntos de interés  $\boldsymbol{\varepsilon}_k^j = (\boldsymbol{\varepsilon}_1^j, \boldsymbol{\varepsilon}_2^j, \boldsymbol{\varepsilon}_3^j, \boldsymbol{\varepsilon}_4^j)$ , en una configuración inicial y final, todo ello junto al modelado de un robot Puma 560.

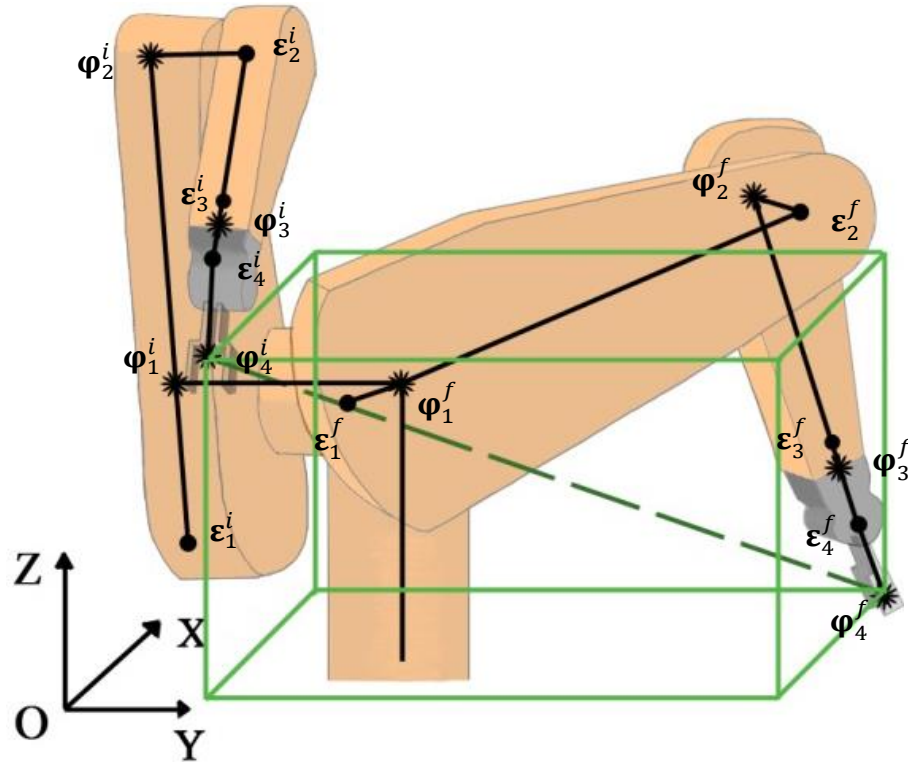


Figura 1. Modelado alámbrico del robot con espacio de trabajo y puntos significativos y de interés.

## 2.2. Problema cinemático

En esta sección se explica el análisis de la cinemática del sistema. La cinemática se define como parte de la mecánica que estudia el movimiento en el espacio sin tener en cuenta las fuerzas que lo producen. Llevado al ámbito de la robótica, consiste en determinar las relaciones matemáticas existentes entre las coordenadas articulares que definen la configuración de un sistema robótico y las coordenadas cartesianas sobre las que se modela el espacio operacional.

Estas relaciones, dependiendo de los datos de partida, se pueden dividir en dos tipos: problemas cinemáticos directos y problemas cinemáticos inversos. Así, si se conoce la configuración del robot, y se quiere determinar el punto sobre el que se sitúa su manipulador, se plantea un problema cinemático directo. En sentido contrario, si lo que se quiere es

determinar la configuración a partir de la posición del manipulador del robot, se aborda como un problema cinemático inverso.

## 2.2.1. Introducción a la cinemática

Un conjunto de cuerpos rígidos, conectados por una serie de articulaciones, determina una cadena cinemática. A los cuerpos rígidos, conectados mediante las articulaciones en estas cadenas cinemáticas, se les denominan *eslabones*. A su vez, las articulaciones suelen ser del tipo prismáticas (puede realizar una traslación en el propio eje de la articulación) o rotacionales (puede girar alrededor del eje de la articulación). Existen otros tipos como cilíndricas (puede rotar y trasladarse), planares (puede desplazarse en un plano) o esféricas (combinan tres giros en tres direcciones). El conjunto de todas las direcciones sobre las que se pueden realizar desplazamientos o giros se le denomina GDL.

Es posible expresar un determinado número de GDL de una articulación como un conjunto de articulaciones que presentan únicamente un GDL y, a su vez, están conectadas por  $(\text{GDL} - 1)$  eslabones. De modo que es posible simplificar cualquier conjunto de cadenas cinemáticas como una combinación de articulaciones que presentan un único grado de libertad, siendo la primera de ellas la denominada base de la cadena y la última el manipulador.

Como se ha explicado anteriormente, una cadena cinemática se puede expresar de numerosas formas, pero puede descomponerse en cadenas más sencillas. En este trabajo se emplea la notación modificada de Denavit y Hartenberg (m-DH), presentada por (Craig 2005).

Este procedimiento permite definir las transformaciones relativas entre los eslabones con únicamente cuatro parámetros, ver Figura 2.

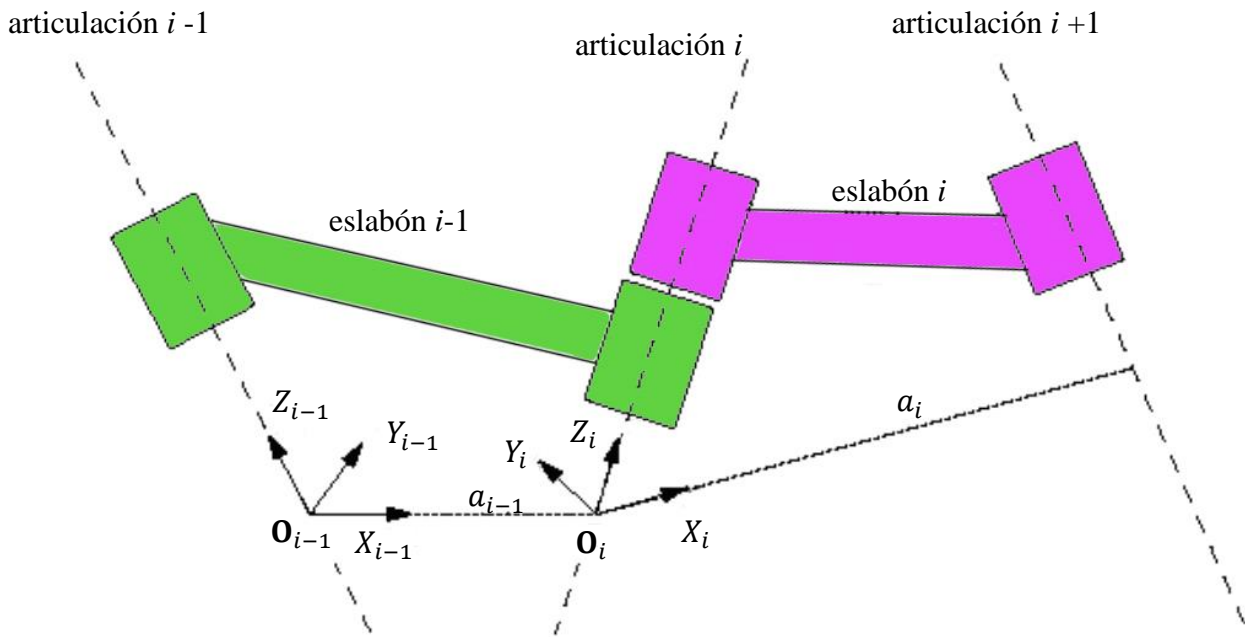


Figura 2. Notación modificada de Denavit y Hartenberg.

- Ángulo  $\theta_i$ : ángulo entre  $X_{i-1}$  y  $X_i$  girando alrededor de  $Z_i$ .
- Distancia  $d_i$ : distancia entre  $X_{i-1}$  hasta  $X_i$  medida a lo largo de  $Z_i$ .
- Distancia  $a_{i-1}$ : distancia desde  $Z_{i-1}$  hasta  $Z_i$  medida a lo largo de  $X_{i-1}$ .
- Ángulo  $\alpha_{i-1}$ : ángulo que hay que girar  $Z_{i-1}$  para alcanzar a  $Z_i$ , girando alrededor de  $X_{i-1}$ .

## 2.2.2. Cinemática directa

La cinemática directa es la metodología encargada de obtener la posición y orientación del manipulador, a partir un conjunto de ángulos entre las articulaciones del robot. Usando los parámetros definidos con m-DH, la matriz de transformación  ${}^{i-1}_i\mathbf{T}$ , que cambia un vector del eslabón i-1 al i, puede expresarse de la siguiente forma:

$${}^{i-1}_i\mathbf{T} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & d_i\sin(\alpha_i) \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & -d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

De esta matriz se pueden extraer:

- La matriz de rotación, que muestra la orientación relativa:

$${}^{i-1}\mathbf{R}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) \end{bmatrix} \quad (2.2)$$

- La matriz de posición, que indica la posición relativa:

$${}^{i-1}\vec{\mathbf{r}}_{\mathbf{0}_{i-1},\mathbf{0}_i} = \begin{bmatrix} a_i \\ d_i\sin(\alpha_i) \\ -d_i\cos(\alpha_{i-1}) \end{bmatrix} \quad (2.3)$$

### 2.2.3. Aplicación de la cinemática directa en Puma un 560

A continuación, se detalla el modelo del Puma 560, que cuenta con 6 GDL. En la Figura 3, se aprecia como todas las articulaciones son rotacionales.

Para determinar el vector de posición del manipulador con respecto a la base, se aplican las ecuaciones (2.2) y (2.3), desde la base hasta el manipulador:

$${}^0\vec{\mathbf{r}}_{\mathbf{0}_0,\mathbf{p}} = {}^0\vec{\mathbf{r}}_{\mathbf{0}_0,\mathbf{0}_i} + {}^0\mathbf{R}_i {}^i\vec{\mathbf{r}}_{\mathbf{0}_{i-1},\mathbf{p}_i} \quad (2.4)$$

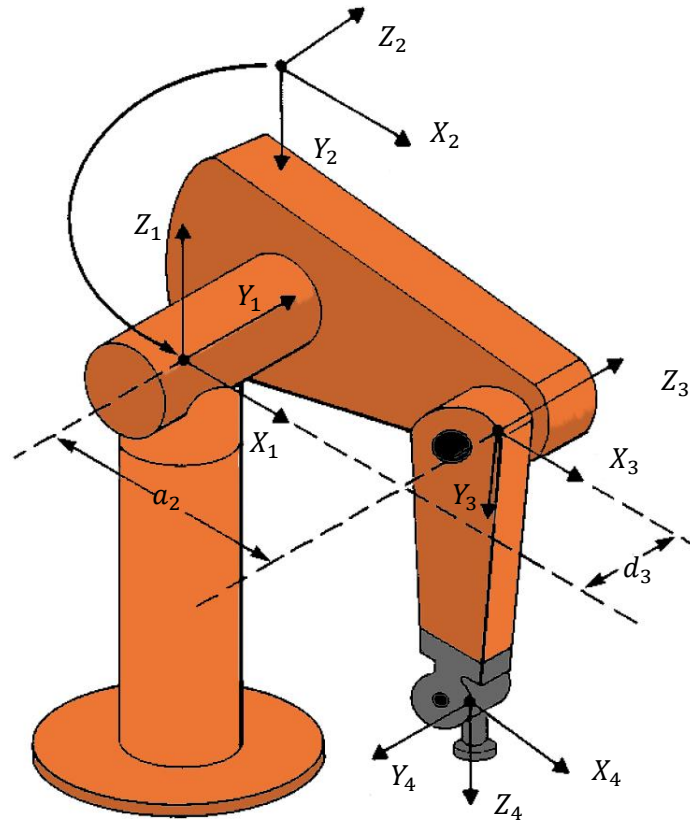


Figura 3. Articulaciones y ejes del Puma 560.

## 2.2.4. Cinemática inversa

El problema de la cinemática inversa, como se ha mencionado antes, consiste en determinar las posibles configuraciones que sitúen al manipulador en un punto, conocida su posición de partida. La cinemática inversa resulta más compleja de determinar que la directa, dado que para una posición concreta pueden existir múltiples soluciones algebraicas.

Las distintas estrategias que solucionan problemas cinemáticos se pueden dividir en dos clases: soluciones de forma cerrada (analíticas) y soluciones numéricas. Las soluciones numéricas son generalmente mucho más lentas que las de solución de forma cerrada, debido a la naturaleza iterativa que presentan. Las soluciones analíticas se basan en la resolución de un polinomio de hasta grado 4, es decir, no se requiere de cálculos iterativos para llegar a una solución. Estos a su vez se subdividen en otros dos métodos para resolver el cálculo cinemático inverso: geométricos y algebraicos. En este proyecto se aplicará el método

algebraico explicado en (Craig 2005) para la resolución de la cinemática inversa para el Puma 560.

## 2.2.5. Aplicación de la cinemática inversa en un Puma 560

A continuación se muestra la resolución del problema cinemático inverso para el robot Puma 560. Toda la formulación se ha planteado siguiendo los modelos que se indican en (Craig 2005).

$${}^0_6\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_X \\ r_{21} & r_{22} & r_{23} & P_Y \\ r_{31} & r_{32} & r_{33} & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0_1\mathbf{T}(\theta_1){}_1^2\mathbf{T}(\theta_2){}_2^3\mathbf{T}(\theta_3){}_3^4\mathbf{T}(\theta_4){}_4^5\mathbf{T}(\theta_5){}_5^6\mathbf{T}(\theta_6) \quad (2.5)$$

Al igual que en la cinemática directa, se puede extraer el vector de posición y la matriz de rotación.

- La matriz de rotación es:  $\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$  con ella se puede obtener la orientación del manipulador.
- El vector de posición es:  $\begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix}$  desde el manipulador hasta la base del robot.

Estos parámetros son:

$$r_{11} = c_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] + s_1(s_4c_5c_6 - c_4s_6) \quad (2.6)$$

$$r_{21} = s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] + c_1(s_4c_5c_6 - c_4s_6) \quad (2.7)$$

$$r_{31} = -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6 \quad (2.8)$$

$$r_{12} = c_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] + s_1(c_4c_6 - s_4c_5s_6) \quad (2.9)$$



$$r_{22} = s_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] - c_1(c_4c_6 - s_4c_5s_6) \quad (2.10)$$

$$r_{32} = -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6 \quad (2.11)$$

$$r_{13} = -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5 \quad (2.12)$$

$$r_{23} = -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5 \quad (2.13)$$

$$r_{33} = s_{23}c_4s_5 - c_{23}c_5 \quad (2.14)$$

$$P_X = c_1[a_2c_2 + a_3c_{23} - d_4s_{23}] - d_3s_1 \quad (2.15)$$

$$P_Y = s_1[a_2c_2 + a_3c_{23} - d_4s_{23}] + d_3c_1 \quad (2.16)$$

$$P_Z = -a_3s_{23} - a_2s_2 - d_4c_{23} \quad (2.17)$$

Siendo:

$$c_i = \cos(q_i) \quad (2.18)$$

$$s_i = \sin(q_i) \quad (2.19)$$

$$c_{ij} = \cos(q_i + q_j) \quad (2.20)$$

$$s_{ij} = \sin(q_i + q_j) \quad (2.21)$$

Por otro lado, los valores de  $q_i$  se calculan con las siguientes expresiones:

$$q_1 = \text{Atan2}(P_X, P_Y) - \text{Atan2}\left(d_3 \pm \sqrt{(P_X^2 + P_Y^2 - d_3^2)}\right) \quad (2.22)$$

Para  $q_3$  es necesario un cálculo intermedio haciendo uso de:

$$K = \frac{P_X^2 + P_Y^2 + P_Z^2 - a_2^2 - a_3^2 - d_3^2 - a_4^2}{2a_2} \quad (2.23)$$

$$q_3 = \text{Atan2}(a_3, d_4) - \text{Atan2}\left(K \pm \sqrt{(a_3^2 + d_4^2 - K^2)}\right) \quad (2.24)$$

Para  $q_2$ , es necesario el uso de  $q_{23}$  de forma que:

$$q_2 = q_{23} - q_3 \quad (2.25)$$

$$q_{23} = \text{Atan2}\left[\begin{array}{l} (-a_3 - a_2 c_3)P_Z - (c_1 P_X + s_1 P_Y)(d_4 - a_2 s_3), \\ (a_2 s_3 - d_4)P_Z - (a_3 - a_2 c_3)(c_1 P_X + s_1 P_Y) \end{array}\right] \quad (2.26)$$

El cálculo de  $q_4$  es directo con la fórmula:

$$q_4 = \text{Atan2}(-r_{13}s_1 + r_{23}c_1, -r_{13}c_1c_{23} - r_{23}s_1c_{23} + r_{33}s_{23}) \quad (2.27)$$

El cálculo de  $q_5$  se deduce mediante:

$$q_5 = \text{Atan2}(s_5, c_5) \quad (2.28)$$

Siendo  $s_5$  y  $c_5$ :

$$s_5 = -[(r_{13}c_{23}c_4 + s_1s_4) + r_{23}(s_1c_{23}c_4 - c_1s_4) - r_{33}(s_{23}c_4)] \quad (2.29)$$

$$c_5 = r_{13}(-c_1s_{23}) + r_{23}(-s_1s_{23}) - r_{33}(-c_{23}) \quad (2.30)$$

Por último, se obtiene  $q_6$ :

$$q_6 = \text{Atan2}(s_6, c_6) \quad (2.31)$$

Siendo  $s_6$  y  $c_6$ :

$$s_6 = -r_{11}(c_1 c_{23} s_4 - s_1 c_4) - r_{21}(s_1 c_{23} s_4 - c_1 c_4) + r_{31}(s_{23} s_4) \quad (2.32)$$

$$s_5 = r_{11}[(c_1 c_{23} c_4 + s_1 s_4)c_5 - c_1 s_{23} s_5] + r_{21}[(s_1 c_{23} c_4 - c_1 s_4)c_5 - s_1 s_{23} s_5] - r_{31}(s_{23} c_4 c_5 + c_{23} s_5) \quad (2.33)$$

## 2.3. Modelado del entorno

En el modelado del entorno podemos distinguir dos ámbitos: el del espacio de trabajo y el de los obstáculos. Ambos se encuentran definidos en coordenadas cartesianas.

### 2.3.1. Modelado del espacio de trabajo

El espacio de trabajo está definido con un sistema de coordenadas cartesianas sobre el que se sitúa un prisma que define sus límites. Una de sus diagonales principales tiene como extremos el punto inicial del que se parte y el punto objetivo. Este prisma está compuesto por otros elementales que se construyen a partir del paso del robot en cada dirección  $X$ ,  $Y$  y  $Z$ , de la siguiente forma:

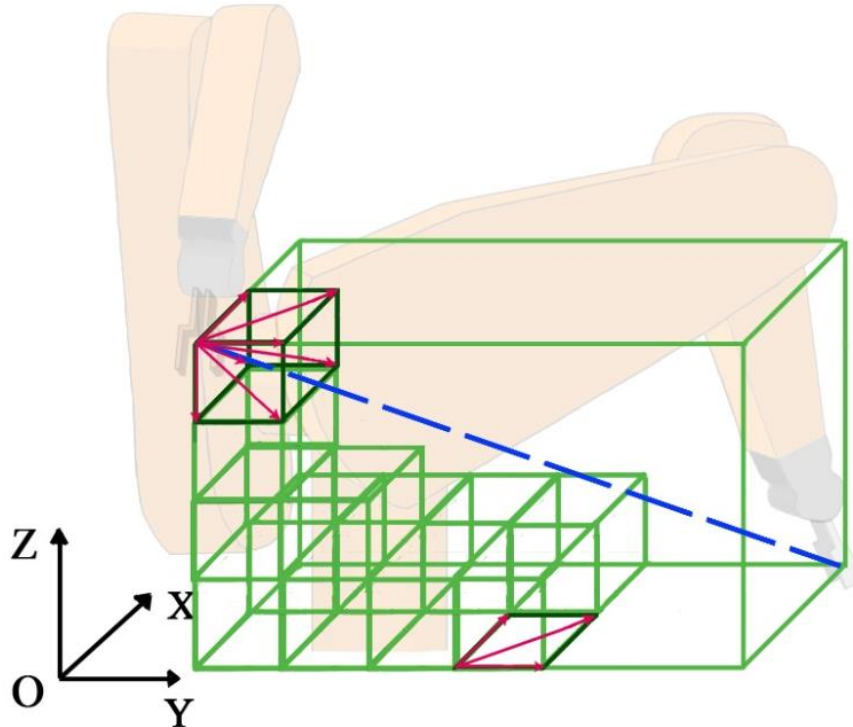
$$\Delta x = \frac{\varphi_{nx}^f - \varphi_{nx}^i}{P_{ts_x} - 1}; \text{ donde } P_{ts_x} = 1 + \text{redon}\left(\frac{|\varphi_{nx}^f - \varphi_{nx}^i|}{D}\right) \quad (2.34)$$

$$\Delta y = \frac{\varphi_{ny}^f - \varphi_{ny}^i}{P_{ts_y} - 1}; \text{ donde } P_{ts_y} = 1 + \text{redon}\left(\frac{|\varphi_{ny}^f - \varphi_{ny}^i|}{D}\right) \quad (2.35)$$

$$\Delta z = \frac{\varphi_{nz}^f - \varphi_{nz}^i}{P_{ts_z} - 1}; \text{ donde } P_{ts_z} = 1 + \text{redon}\left(\frac{|\varphi_{nz}^f - \varphi_{nz}^i|}{D}\right) \quad (2.36)$$

Donde  $\varphi_n^f - \varphi_n^i$  en cada dirección  $X$ ,  $Y$  y  $Z$ , es la distancia entre las posiciones del elemento final de las configuraciones inicial y final del robot. En el caso del Puma 560,  $n$  tiene los valores de 1 hasta 4. La variable  $D$  toma el valor mínimo entre los tamaños de los obstáculos y diámetros de los brazos del robot. Es decir, si el diámetro de un brazo del robot es el de menor tamaño,  $D$  equivaldrá a su valor. La función  $\text{redon}(\text{número})$  calcula el valor integral más pequeño que no sea menor del número. Por otra parte,  $(P_{ts_x} - 1, P_{ts_y} - 1, P_{ts_z} - 1)$  corresponden con el número de pasos y por tanto, de puntos en cada dirección del prisma construido en el espacio de trabajo.

Se forma así, dentro del espacio operativo, una estructura de nodos que simula los posibles puntos por los que se pueden generar rutas -desde el inicial hasta el final-, como se aprecia en la Figura 4.



**Figura 4.** Ejemplo espacio de trabajo. Un prisma, donde los puntos extremos de la diagonal principal son el punto de inicio y el de destino. Desde un nodo se puede acceder como máximo a otros siete mediante sus arcos.

A partir de un mismo punto se pueden escoger hasta 7 arcos distintos, teniendo en cuenta que sólo se puede avanzar por ellos hacia el nodo de destino y condicionados por la distancia mínima que puede moverse el robot en cada una de las direcciones.

En la Figura 4 se observa una pequeña distribución de cómo se va discretizando el espacio. Ésta se aplicaría a todo el prisma pero, para facilitar la visualización, sólo se discretiza una parte del espacio.

### **2.3.2. Modelado de obstáculos**

Los obstáculos imposibilitan el recorrido por algunas zonas. La inviabilidad se produce no sólo porque algunos nodos se encuentran dentro de los obstáculos, sino también porque no exista ninguna configuración de los brazos del robot que, siendo teóricamente factible, no evite las colisiones. Lo anterior impone que algunos nodos sean inaccesibles dentro de nuestro espacio de trabajo y, consecuentemente, por ellos no se puedan generar “arcos” y por lo tanto, tampoco caminos.

Para definir los modelos de obstáculos, se han generado tres arquetipos sencillos: planos, cilindros y esferas. Utilizando una combinación de estos, es posible generar obstáculos complejos que se puedan asemejar a un amplio conjunto de cuerpos. Dado que el algoritmo se plantea en planificación de caminos globales, los obstáculos serán estáticos. Esto significa que no sufrirán un cambio de orientación o posición una vez definidos. Para todo ello, se ha utilizado la técnica de crecimiento de obstáculos de (Lozano-Pérez and Wesley 1979).

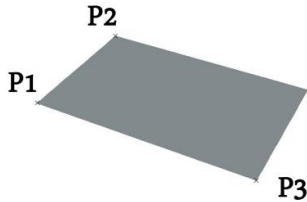
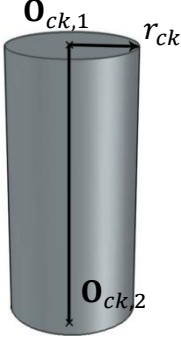
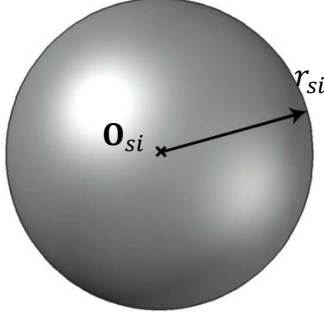
Plano	Cilindro	Esfera
		
$P_1 \rightarrow$ Vértice1 $P_2 \rightarrow$ Vértice2 $P_3 \rightarrow$ Vértice3	$O_{ck,1} \rightarrow$ Centro1 $O_{ck,2} \rightarrow$ Centro2 $r_{ck} \rightarrow$ radio	$O_{si} \rightarrow$ Centro $r_{si} \rightarrow$ radio

Figura 5: Tipos de obstáculos y datos que los definen.

## 2.4. Cálculo de distancias mínimas de colisión

Para desarrollar rutas libres de colisiones es necesario incorporar una formulación que anuncie cuándo cualquier brazo del robot se encuentra impactando con un obstáculo. Es por ello que este capítulo se divide en tres subsecciones, una por cada obstáculo definido anteriormente: esferas, cilindros y planos.

Cada brazo del robot es considerado como un cilindro, de radio  $r_m$  y cuya altura está comprendida entre dos puntos -puntos significativos y/o de interés-. De esta forma, el problema puede reducirse a restringir la distancia entre los cilindros de cada uno de los eslabones del robot para cada uno de los obstáculos.

### 2.4.1. Distancias mínimas con esferas

Considerando la técnica de crecimiento de obstáculos, se ha diseñado una función que calcula la mínima distancia entre los brazos del robot -estimados como segmentos entre los

puntos que le definen- y la esfera. La distancia crítica ha de ser comparada con  $r_{si} + r_n$  y su valor dependerá de  $\overrightarrow{P_n \epsilon_n}$ :

$$\overrightarrow{P_n \epsilon_n} = \text{proyección}(\overrightarrow{\epsilon_n O_{si}}, \overrightarrow{\epsilon_n \varphi_n}) \quad (2.37)$$

De forma que si  $\|\overrightarrow{P_n \epsilon_n}\| > \|\overrightarrow{\epsilon_n \varphi_n}\|$ , el valor crítico será  $\|\overrightarrow{\epsilon_n O_{si}}\|$ . Si no lo es, se comprobará si es menor que 0. Si resulta ser negativo, el valor crítico será  $\|\overrightarrow{P_n O_{si}}\|$  y si no lo es, será  $\|\overrightarrow{\epsilon_n O_{si}}\|$ . Esto se explica en el siguiente diagrama de flujo:

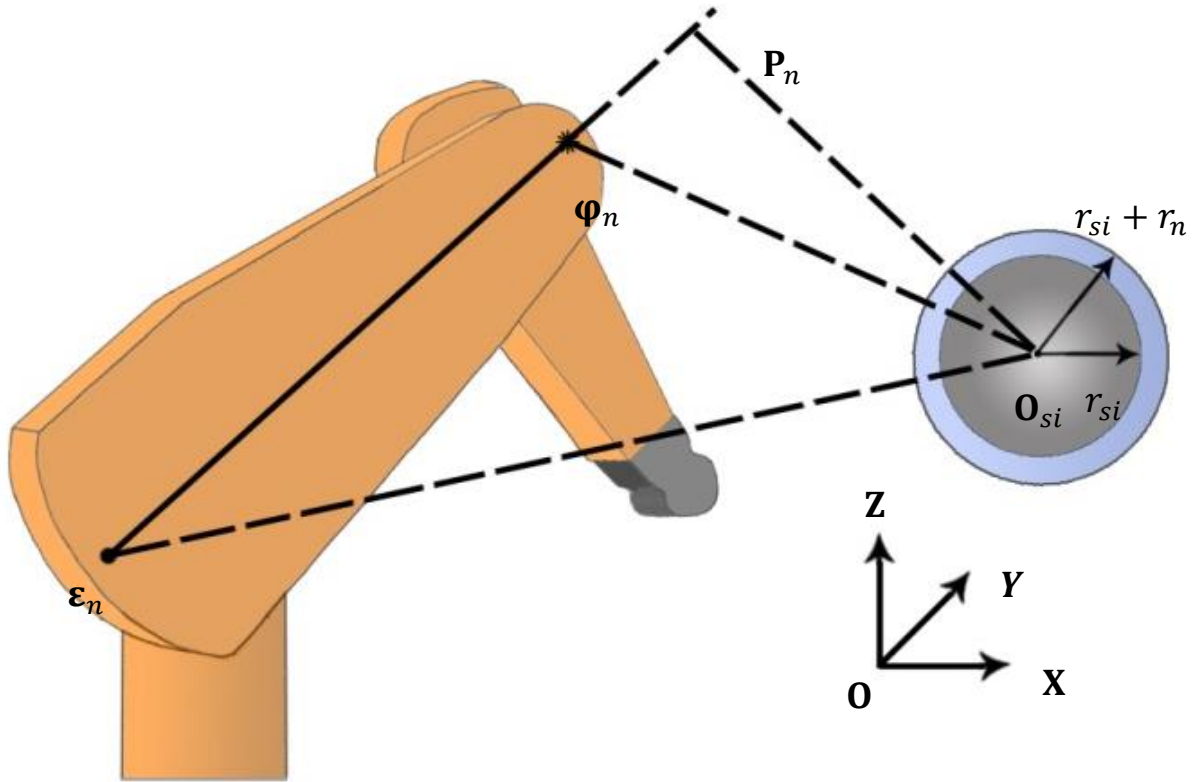


Figura 6. Distancias mínimas con esferas.

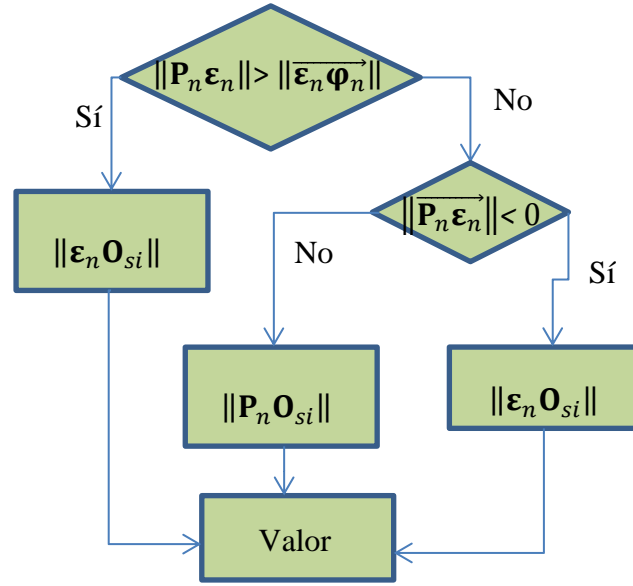


Figura 7. Diagrama de flujo colisiones con esferas.

## 2.4.2. Distancias mínimas con cilindros

Del mismo modo que en el caso de la esfera, el objetivo es determinar la distancia mínima entre el robot y el cilindro, de forma que podamos determinar si existe realmente una colisión. De modo que pueda ser un extremo del brazo el que se sitúe más cerca del cilindro o cualquier punto contenido entre los extremos.

Para ello expresamos, como funciones lineales, los siguientes puntos:

$$\mathbf{P}_n = \boldsymbol{\varepsilon}_n + (\boldsymbol{\varphi}_n - \boldsymbol{\varepsilon}_n) * x_1 \quad (2.38)$$

$$\mathbf{P}_k = \boldsymbol{\varepsilon}_n + (\boldsymbol{\varphi}_n - \boldsymbol{\varepsilon}_n) * x_1 \quad (2.39)$$

$$\mathbf{P}_k - \mathbf{P}_n = \mathbf{O}_{ck,1} + (\mathbf{O}_{ck,2} - \mathbf{O}_{ck,1}) * x_2 - \boldsymbol{\varepsilon}_n - (\boldsymbol{\varphi}_n - \boldsymbol{\varepsilon}_n) * x_1 \quad (2.40)$$



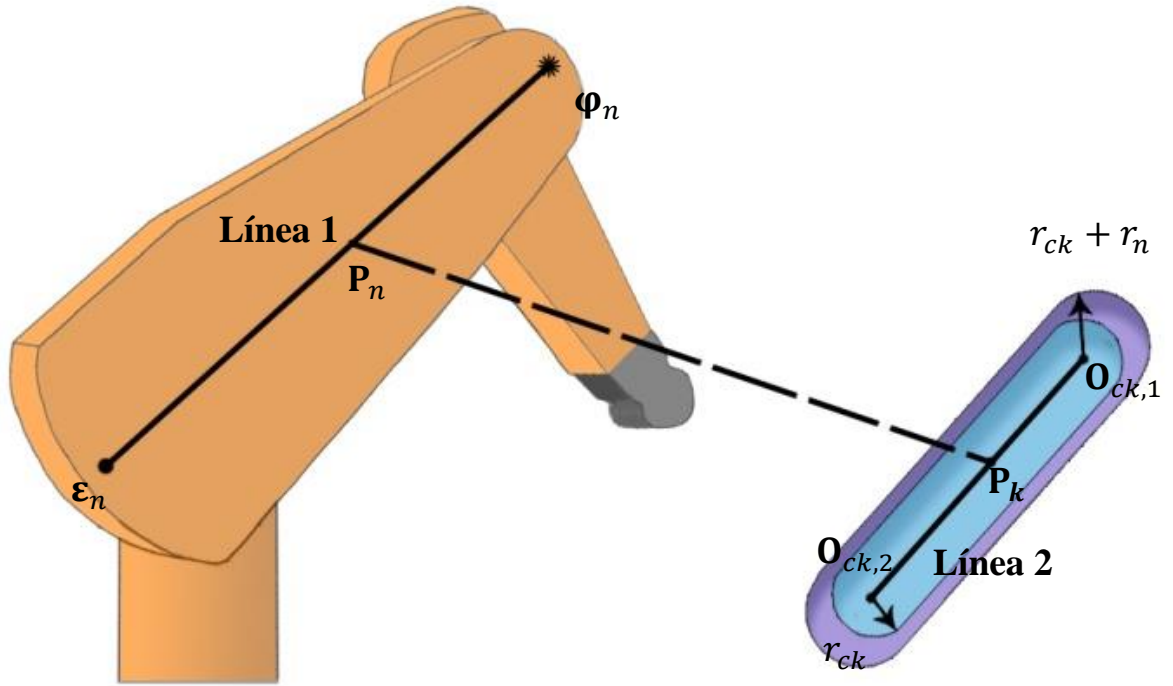


Figura 8. Distancias mínimas con cilindros.

Dado por condiciones de perpendicularidad simultánea a la Línea 1 y 2:

$$\overrightarrow{\epsilon_n \phi_n} * \overrightarrow{P_n P_k} = 0 \quad (2.41)$$

$$\overrightarrow{O_{ck,1} O_{ck,2}} * \overrightarrow{P_n P_k} = 0 \quad (2.42)$$

Tenemos:

$$\begin{aligned} & \begin{bmatrix} -\overrightarrow{\epsilon_n \phi_n} * \overrightarrow{\epsilon_n \phi_n} & \overrightarrow{\epsilon_n \phi_n} * \overrightarrow{O_{ck,1} O_{ck,2}} \\ -\overrightarrow{\epsilon_n \phi_n} * \overrightarrow{O_{ck,1} O_{ck,2}} & \overrightarrow{O_{ck,1} O_{ck,2}} * \overrightarrow{O_{ck,1} O_{ck,2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} \overrightarrow{\epsilon_n \phi_n} * (\overrightarrow{\epsilon_n} * \overrightarrow{O_{ck,1}}) \\ \overrightarrow{O_{ck,1} O_{ck,2}} * (\overrightarrow{\epsilon_n} * \overrightarrow{O_{ck,1}}) \end{bmatrix} \end{aligned} \quad (2.43)$$

O lo que es lo mismo:

$$[\mathbf{m}] * \mathbf{x} = \begin{bmatrix} A \\ B \end{bmatrix} \quad (2.44)$$

Despejando  $x_1$  y  $x_2$ :

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{\overrightarrow{\mathbf{O}_{ck,1} \mathbf{O}_{ck,2}} * \overrightarrow{\mathbf{O}_{ck,1} \mathbf{O}_{ck,2}}}{|\mathbf{m}|} A - \frac{\overrightarrow{\boldsymbol{\varepsilon}_n \boldsymbol{\varphi}_n} * \overrightarrow{\mathbf{O}_{ck,1} \mathbf{O}_{ck,2}}}{|\mathbf{m}|} B \\ \frac{\overrightarrow{\boldsymbol{\varepsilon}_n \boldsymbol{\varphi}_n} * \overrightarrow{\mathbf{O}_{ck,1} \mathbf{O}_{ck,2}}}{|\mathbf{m}|} A - \frac{\overrightarrow{\boldsymbol{\varepsilon}_n \boldsymbol{\varphi}_n} * \overrightarrow{\boldsymbol{\varepsilon}_n \boldsymbol{\varphi}_n}}{|\mathbf{m}|} B \end{bmatrix} \quad (2.45)$$

Dado que ahora  $x_1$  y  $x_2$  representan toda la línea entre los puntos  $\overrightarrow{\boldsymbol{\varepsilon}_n \boldsymbol{\varphi}_n}$  y  $\overrightarrow{\mathbf{O}_{ck,1} \mathbf{O}_{ck,2}}$  es necesario definir nuevos parámetros para acotar los segmentos:

$$x_{11} = \begin{cases} 0 & , \text{si } x_1 < 0 \\ 1 & , \text{si } x_1 > 0 \\ x_1 & , \text{si } 0 \leq x_2 \leq 1 \end{cases} \quad (2.46)$$

$$x_{22} = \begin{cases} 0 & , \text{si } x_2 < 0 \\ 1 & , \text{si } x_2 > 0 \\ x_1 & , \text{si } 0 \leq x_2 \leq 1 \end{cases} \quad (2.47)$$

De esta forma podemos concluir con el diagrama de flujo que se muestra en la Figura 9 cómo se determinará la distancia crítica.

Siendo

$$\overrightarrow{\mathbf{d}_{k,1}} = \overrightarrow{\mathbf{P}_n \mathbf{P}_k} \quad (2.48)$$

Finalmente, se le resta  $r_{ck} + r_n$  al valor de  $\|\mathbf{d}_k\|$  y se obtiene la mínima distancia real.

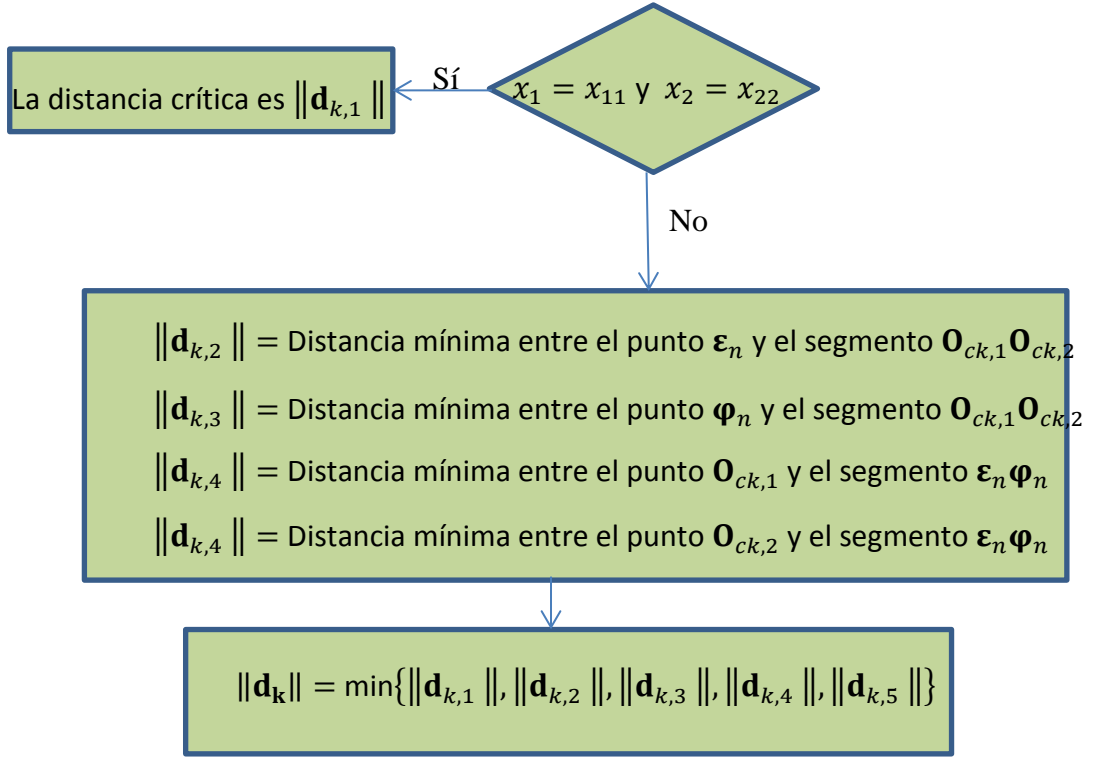


Figura 9. Diagrama de flujo colisiones con cilindros.

### 2.4.3. Distancias mínimas con planos

Para determinar la distancia mínima entre un brazo del robot y un plano, es necesario comprobar qué se encuentra más cercano a él, uno de los extremos o un punto del segmento entre ambos.

Para determinar las proyecciones de los puntos es necesario calcular los vectores unitarios que definen las direcciones del plano  $\vec{r}$ ,  $\vec{s}$  y  $\vec{n}$ :

$$\vec{r} = \frac{\mathbf{P}_2 - \mathbf{P}_1}{\|\mathbf{P}_2 - \mathbf{P}_1\|} \quad (2.49)$$

$$\vec{n} = \vec{r} \times \frac{\mathbf{P}_3 - \mathbf{P}_1}{\|\mathbf{P}_3 - \mathbf{P}_1\|} \quad (2.50)$$

$$\vec{s} = \vec{n} \times \vec{r} \quad (2.51)$$

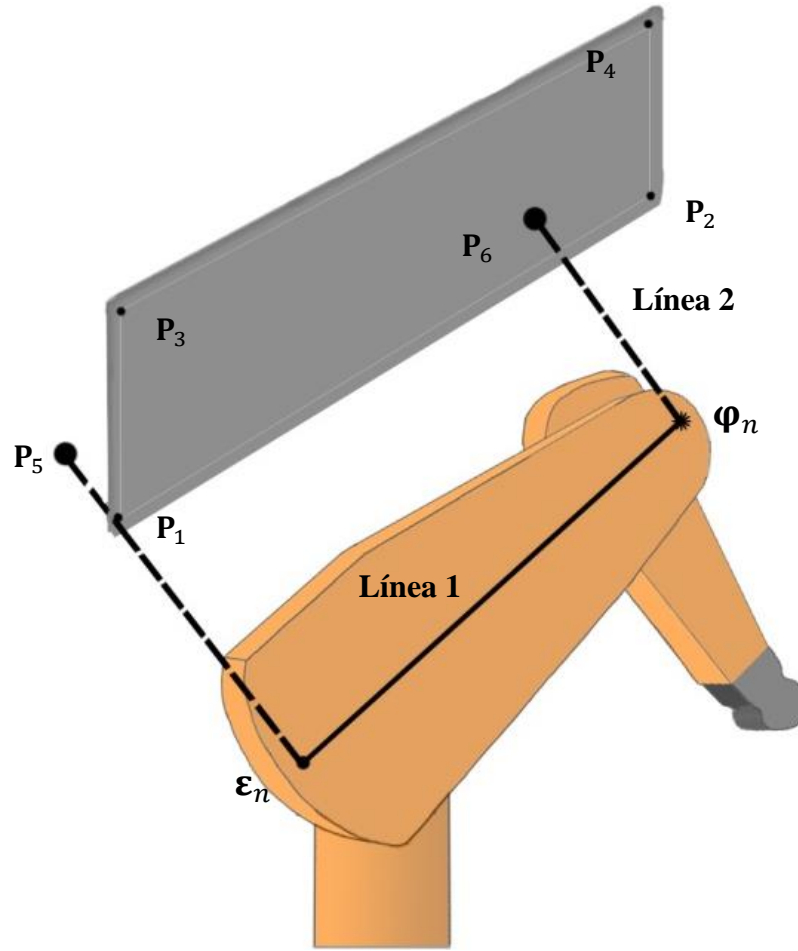


Figura 10. Distancias mínimas con planos.

De esta forma:

$$\mathbf{P}_5 = (\boldsymbol{\varepsilon}_n - \mathbf{P}_1) * \vec{\mathbf{r}} + (\boldsymbol{\varepsilon}_n - \mathbf{P}_1) * \vec{\mathbf{s}} + \mathbf{P}_1 \quad (2.52)$$

$$\mathbf{P}_6 = (\boldsymbol{\varphi}_n - \mathbf{P}_1) * \vec{\mathbf{r}} + (\boldsymbol{\varphi}_n - \mathbf{P}_1) * \vec{\mathbf{s}} + \mathbf{P}_1 \quad (2.53)$$

En caso de que el plano sea infinito, la distancia mínima es:

$$\|\mathbf{d}_{n,1}\| = \min\{\|\boldsymbol{\varepsilon}_n - \mathbf{P}_5\|, \|\boldsymbol{\varphi}_n - \mathbf{P}_6\|\} \quad (2.54)$$

Para determinar si el punto  $\mathbf{P}_6$  se encuentra dentro del plano tendrá que cumplirse:

$$\begin{aligned} ((\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_6 - \mathbf{P}_1)) * \vec{\mathbf{n}} &> 0, \\ ((\mathbf{P}_4 - \mathbf{P}_2) \times (\mathbf{P}_6 - \mathbf{P}_2)) * \vec{\mathbf{n}} &> 0, \\ ((\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_6 - \mathbf{P}_4)) * \vec{\mathbf{n}} &> 0, \\ ((\mathbf{P}_1 - \mathbf{P}_3) \times (\mathbf{P}_6 - \mathbf{P}_3)) * \vec{\mathbf{n}} &> 0, \end{aligned} \tag{2.55}$$

De la misma manera para el punto  $\mathbf{P}_5$ :

$$\begin{aligned} ((\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_5 - \mathbf{P}_1)) * \vec{\mathbf{n}} &> 0, \\ ((\mathbf{P}_4 - \mathbf{P}_2) \times (\mathbf{P}_5 - \mathbf{P}_2)) * \vec{\mathbf{n}} &> 0, \\ ((\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_5 - \mathbf{P}_4)) * \vec{\mathbf{n}} &> 0, \\ ((\mathbf{P}_1 - \mathbf{P}_3) \times (\mathbf{P}_5 - \mathbf{P}_3)) * \vec{\mathbf{n}} &> 0, \end{aligned} \tag{2.56}$$

Si se llega a la conclusión de que existe una proyección para un punto dentro del plano, habrá que confirmar que los extremos de los brazos  $\boldsymbol{\varepsilon}_n$  y  $\boldsymbol{\varphi}_n$  se encuentran en la misma cara del plano, de forma que cumplan:

$$\frac{(\boldsymbol{\varepsilon}_n - \mathbf{P}_5)}{\|\boldsymbol{\varepsilon}_n - \mathbf{P}_5\|} = \frac{(\boldsymbol{\varphi}_n - \mathbf{P}_6)}{\|\boldsymbol{\varphi}_n - \mathbf{P}_6\|} \tag{2.57}$$

Si se resuelve que sólo un punto tiene proyección, entonces habrá que calcular la distancia mínima entre dos segmentos. Ésta será  $\|\mathbf{d}_{n,1}\|$  si el brazo no se encuentra sobre la cara del plano. Si el brazo se encuentra sobre la cara del plano, entonces será el mínimo entre:

$$\begin{aligned} \|\mathbf{d}_{n,2}\| &= \text{distancia mínima entre los segmentos } \boldsymbol{\varepsilon}_n\boldsymbol{\varphi}_n \text{ y } \mathbf{P}_1\mathbf{P}_2 \\ \|\mathbf{d}_{n,3}\| &= \text{distancia mínima entre los segmentos } \boldsymbol{\varepsilon}_n\boldsymbol{\varphi}_n \text{ y } \mathbf{P}_2\mathbf{P}_3 \\ \|\mathbf{d}_{n,4}\| &= \text{distancia mínima entre los segmentos } \boldsymbol{\varepsilon}_n\boldsymbol{\varphi}_n \text{ y } \mathbf{P}_3\mathbf{P}_4 \\ \|\mathbf{d}_{n,5}\| &= \text{distancia mínima entre los segmentos } \boldsymbol{\varepsilon}_n\boldsymbol{\varphi}_n \text{ y } \mathbf{P}_4\mathbf{P}_1 \end{aligned} \tag{2.58}$$

De forma que tenemos el siguiente diagrama de flujo:

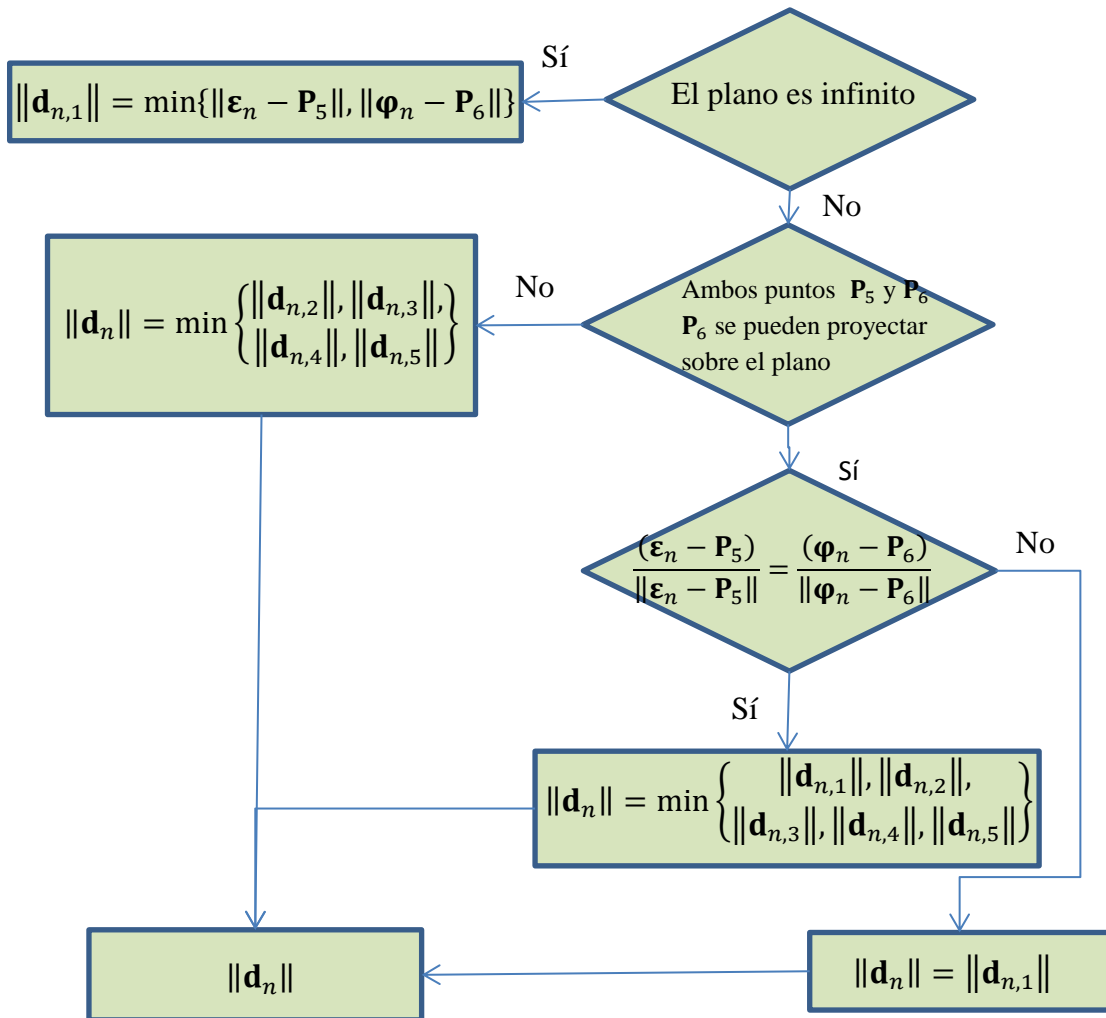


Figura 11. Diagrama de flujo colisiones con planos.

# Capítulo 3

## El ACO en la planificación de caminos

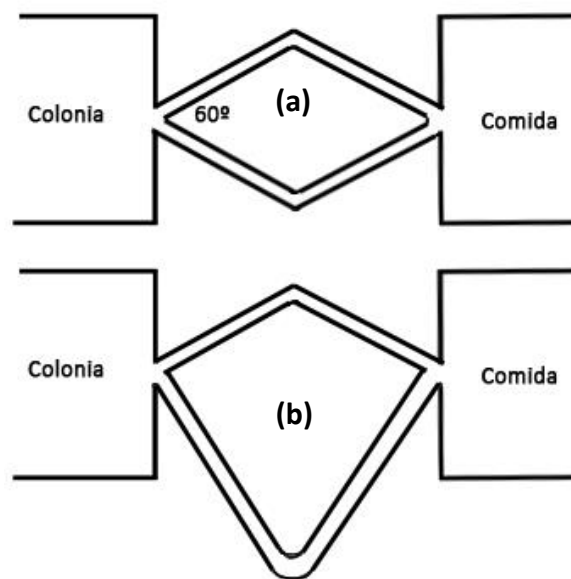
El ACO planteado por Dorigo en 1992, es un algoritmo heurístico de búsqueda que usando la vecindad de los nodos del entorno, determina una ruta para problemas combinatorios complejos. A continuación, se explicará en qué se basa el algoritmo, cómo se puede implementar, su funcionamiento, sus modificaciones generalizadas, un pequeño caso práctico y el método de obtención de caminos que hemos desarrollado con él.

### 3.1. Perspectiva general del ACO

La teoría del algoritmo de las hormigas se basa en el comportamiento que siguen éstas en sus desplazamientos. Es decir:

- Siguen unas rutas aleatorias hasta encontrar comida y, de regreso a la colonia, se deja un rastro de feromona.
- Este rastro puede ser detectado por otra hormiga, que lo reforzará si ha encontrado alimento.
- La feromona se va evaporando y, por lo tanto, los caminos van perdiendo todo su rastro si no son frecuentados.

Esto se ha demostrado con numerosos experimentos, tales como el realizado en (Goss et al. 1989). En él se tienen dos colonias distintas separadas por distintos arcos hacia la comida:



**Figura 12. Experimento “Self-organized Shortcuts in the Argentine Ant”**

La colonia (a) tiene dos caminos de la misma longitud posibles hacia la comida, mientras que la (b) tiene un camino más largo que otro. Los resultados del experimento muestran que en (a) la probabilidad de elección de uno de los arcos es la misma a la otra, es decir, las hormigas se desplazan las mismas veces por el arco superior que el inferior. Por el contrario, en la colonia (b) el número de elecciones que se toman en el arco de menor longitud ronda el 90%.



### 3.1.1. Introducción de la optimización

Las bases para transformar el comportamiento que siguen las hormigas a un algoritmo son las siguientes:

1. **Extrapolar las trayectorias:** es necesario modelar un espacio sobre el que puedan realizarse todos los desplazamientos. En modelo hay que considerar hacia donde se puede avanzar desde cada una de las posiciones. Ejemplo de esto es el grafo que se muestra en la Figura 13.

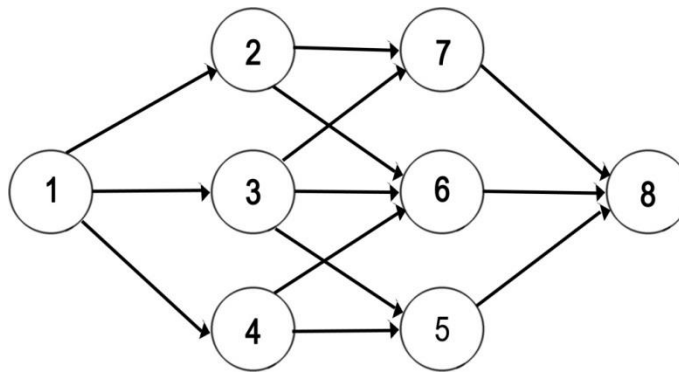


Figura 13. Ejemplo de grafo.

2. **Cuantificar las características que influyen en las decisiones,** de modo que se pueda emplear un modelo que siga sus elecciones. Estas características se pueden dividir en dos tipos:
  - **Información heurística ( $\eta$ ):** se corresponde con la heurística local, su valor no se modifica durante la ejecución del algoritmo. El más utilizado corresponde con la distancia entre nodos, aunque existen otros como la visibilidad de un nodo (número de salidas que tiene un nodo).
  - **Información memorística ( $\tau$ ):** se trata de la información que se puede ir modificando durante la ejecución, y se corresponde con los niveles de feromona depositados.
  -

### 3.1.2. Explicación del algoritmo

El algoritmo puede explicarse como:

1. Se basa en elecciones probabilísticas que toman las hormigas “artificiales”.
  2. Cada hormiga artificial tiene que generar un camino completo por el grafo, es decir, tiene un inicio y un fin, no puede darse por completo un camino que no llegue al destino.
- Estas elecciones probabilísticas siguen el modelo propuesto en la regla de transición:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{j \in J_i^k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} \quad (3.1)$$

Donde  $\alpha$  y  $\beta$  son dos parámetros ajustables,  $i$  es el nodo sobre el que se encuentra la hormiga  $k$ ,  $j$  es el nodo de destino y  $J$  son los nodos alcanzables por la hormiga  $k$  desde el nodo  $i$ .

- La actualización de la feromona, es decir, tanto el incremento como su evaporación, se realiza al final de cada iteración (salida de todas las hormigas de la colonia) sigue el siguiente modelo:

$$\tau_{ij}(t) = (1 - \rho) * \tau_{ij}(t - 1) + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3.2)$$

Donde,  $\rho$  es ratio de evaporación,  $m$  el número de hormigas,  $\Delta\tau_{ij}^k$  cantidad de feromona que la hormiga  $k$  deposita en los arcos por los que pasa (su valor es la inversa de la longitud del circuito).

### 3.1.3. Ejemplo práctico

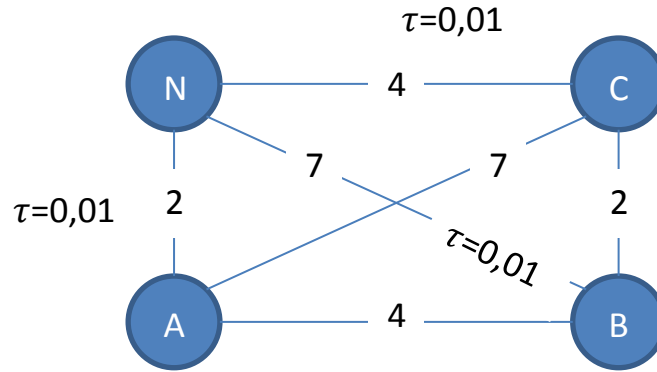
En este ejemplo se mostrará un caso sencillo para la optimización de un problema del tipo viajante (Traveling Salesman Problem). Esta simulación tiene como condiciones: terminar volviendo al inicio y encontrar la ruta más corta pero pasando por todos los nodos y sin repetir el acceso a uno ya visitado.

Para buscar la solución óptima se definen los siguientes parámetros del algoritmo:

**Tabla 1. Parámetros para el ejemplo de TSP.**

$\alpha$	$\beta$	$\rho$	$k$	$\tau_0$
1	1	0,1	2	0,01

Y como entorno se define en la Figura 14. Donde el número situado entre dos nodos corresponde a la distancia que los separa.



**Figura 14. Grafo del TSP**

Si situamos el nodo inicial en N tendremos las siguientes probabilidades para avanzar por el grafo:

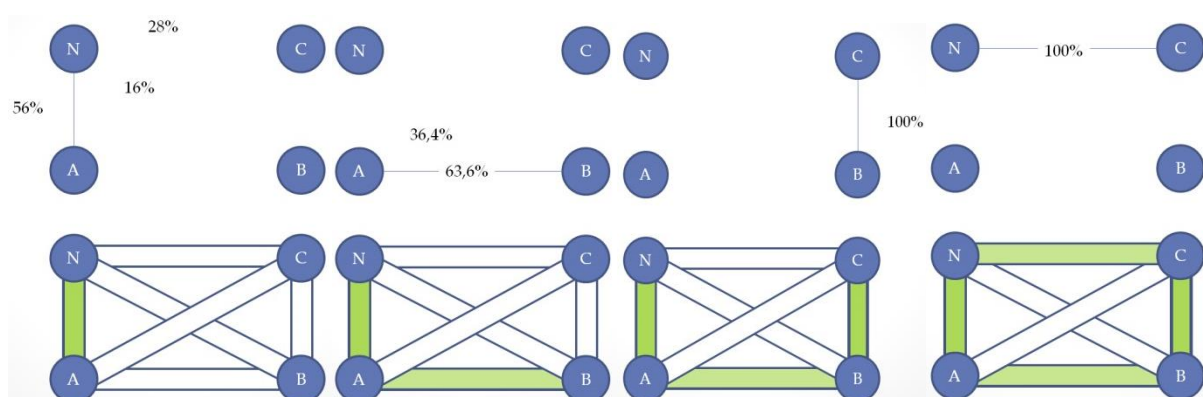
$$P_{N-A} = \frac{(0,01)^{\alpha} \cdot (1/2)^{\beta}}{(0,01)^{\alpha} \cdot (1/2)^{\beta} + (0,01)^{\alpha} \cdot (1/7)^{\beta} + (0,01)^{\alpha} \cdot (1/4)^{\beta}} = 0,56$$

$$P_{N-B} = \frac{(0,01)^{\alpha} \cdot (1/7)^{\beta}}{(0,01)^{\alpha} \cdot (1/2)^{\beta} + (0,01)^{\alpha} \cdot (1/7)^{\beta} + (0,01)^{\alpha} \cdot (1/4)^{\beta}} = 0,16$$

$$P_{N-C} = \frac{(0,01)^{\alpha} \cdot (1/4)^{\beta}}{(0,01)^{\alpha} \cdot (1/2)^{\beta} + (0,01)^{\alpha} \cdot (1/7)^{\beta} + (0,01)^{\alpha} \cdot (1/4)^{\beta}} = 0,28$$

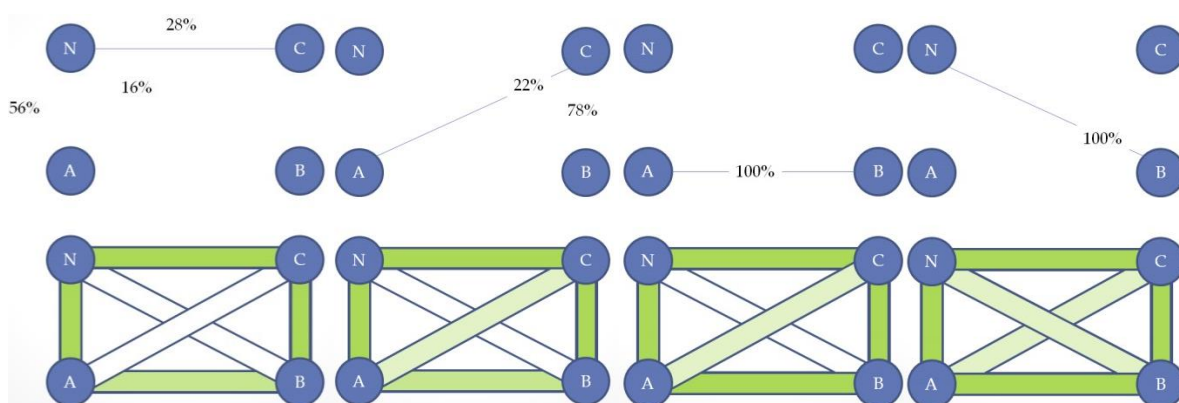
Para continuar, y dado que tenemos dos hormigas ( $k = 2$ ), con una haremos el caso ideal de elegir siempre la mejor posibilidad y con la otra se elegirá algún caso desfavorable. Mientras se puede ir observando como aumentaría la concentración de feromona, teniendo en cuenta, como se ha mencionado antes que este valor sólo se actualizará después de que la segunda hormiga acabe su camino. Por tanto, las decisiones tomadas por la primera no afectarán a las elecciones que toma la segunda hormiga.

#### Hormiga 1:



**Figura 15. TSP realizado por la primera hormiga.**

#### Hormiga 2:



**Figura 16. TSP realizado por la segunda hormiga.**

Cuando la última hormiga ha terminado el recorrido, se actualizan los niveles de feromona como se muestra a continuación.

$$\text{Arco N-A: } \tau_{NA} = (1-0,5)*0,01 + (1/2)=0,505$$

$$\text{Arco A-B: } \tau_{AB} = (1-0,5)*0,01 + [(1/4) + (1/4)]=0,505$$

$$\text{Arco B-C: } \tau_{BC} = (1-0,5)*0,01 + (1/2)=0,505$$

$$\text{Arco N-C: } \tau_{NC} = (1-0,5)*0,01 + [(1/4) + (1/4)]=0,505$$

$$\text{Arco N-B: } \tau_{NB} = (1-0,5)*0,01 + (1/7)=0,148$$

$$\text{Arco A-C: } \tau_{AC} = (1-0,5)*0,01 + (1/7)=0,148$$

Actualizados los niveles, se procedería a ejecutar este mismo proceso tantas veces como iteraciones se definan.

### 3.1.4. Variaciones

Con el fin de optimizar el algoritmo, se han desarrollado numerosas variaciones que modifican algunas de sus características y reglas e incorporar otras nuevas.

#### 3.1.4.1. Sistema de hormigas elitista (EAS)

Se da un mayor peso al mejor camino encontrado hasta el momento, de forma que se modifica la regla de actualización de feromona:

$$\tau_{ij}(t) = (1 - \rho) * \tau_{ij}(t - 1) + \sum_{k=1}^m \Delta\tau_{ij}^k + e * \Delta\tau_{ij}^{bs} \quad (3.3)$$

Siendo:

➤  $e$  = parámetro, número de hormigas elitistas.

$$\text{➤ } \Delta\tau_{ij}^{bs} = \begin{cases} \frac{1}{L^{bs}} , & \text{si el arco (i,j)} \in L_{ij}^{\text{best so far}} \\ 0 , & \text{en otro caso} \end{cases}$$

### 3.1.4.2. Sistema de hormigas MAX-MIN (MMAS)

En esta variación, el valor memorístico está limitado por un máximo y un mínimo, tomándose en el punto de partida el valor máximo. Por otro lado, los rastros de feromona sólo son incrementados por la mejor hormiga, es decir, la influencia de toda la colonia en la exploración se basa únicamente en un individuo, lo que facilita que se converja antes en una solución, pero no es recomendable su uso en entornos muy grandes ya que, al incrementar sólo un camino, es más probable que se produzca un estancamiento en un mínimo local. Otra modificación que presenta este sistema es la de reiniciar los valores memorísticos cuando se alcanza esta situación de mínimo local.

### 3.1.4.3. Sistema de hormigas basado en rankings( $SA_{rank}$ )

En la colonia se establece un ranking de hormigas basándose en la longitud recorrida por cada una. De este modo, cuanto más arriba se situó una hormiga en este ranking, mayor valor memorístico depositará en los arcos. A su vez, las hormigas que no se encuentren dentro de este ranking no podrán incrementar la feromona del entorno.

$$\tau_{ij}(t) = (1 - \rho) * \tau_{ij}(t - 1) + \sum_{k=1}^m (w - \mu) * \Delta\tau_{ij}^k + w * \Delta\tau_{ij}^{bs} \quad (3.4)$$

Siendo:

- $w$  = tamaño del ranking
- $\mu$  = posición que ocupa en el ranking

### 3.1.4.4. Sistema Colonias de Hormigas (ACS)

El valor memorístico se divide en dos tipos: feromona global y feromona local. De este modo, el rastro global sólo lo actualiza la mejor hormiga (como en el caso del MMAS), mientras que el rastro local es actualizado por todas las hormigas.

$$\tau_{i,xi}(t) = (1 - \rho) * \tau_{i,xi}(t - 1) + \rho * \Delta\tau_{i,xi}^{bs} \quad \forall \Delta\tau_{i,xi} \in L^{bs} \quad (3.5)$$

$$\tau_{i,xi}(t) = (1 - \xi) * \tau_{i,xi}(t - 1) + \xi * \tau_{init} \quad (3.6)$$

Siendo  $\xi$  la tasa local de evaporación de feromona

Por otro lado, se modifica la regla de transición entre nodos, como se muestra en la ecuación (3.7)

$$s = \begin{cases} \arg \max_{u \in J_k(r)} ([\tau_{ru}]^\alpha [\eta_{ru}]^\beta), & \text{si } q \leq q_0 \\ S & \text{en otro caso} \end{cases}, \quad (3.7)$$

Siendo:

- $s$  = siguiente nodo al que se desplaza la hormiga  $k$ .
- $q$  = valor aleatorio de una distribución uniforme  $[0,1]$ .
- $q_0 \in [0,1]$  = parámetro probabilístico de escoger otro arco.
- $S$  = nodo seleccionado con la regla de la transición estándar.

Se consigue así aumentar las probabilidades de que se elija el nodo que cuenta con mayor posibilidad por pesos heurísticos y memorísticos.

## 3.2. Obtención de caminos mediante ACO

Antes de poder aplicar el ACO, es necesario desarrollar un entorno donde se disponga de todas posiciones accesibles para el robot, así como de los arcos por los que se puede mover de una posición a otra. En esta sección se explicará el algoritmo implementado que posibilita la generación de todo este espacio operacional.

En primer lugar, se realiza la discretización de un espacio formado por la configuración inicial y final, como se explica en la sección 2.3. Dentro del algoritmo, la discretización se ha definido de dos formas distintas: de forma completa o de forma parcial. La forma completa supone generar íntegramente el espacio operacional, con todos los nodos que definen el prisma de la Figura 4. De esta forma, se originan todos los caminos que definen el conjunto total de las posibles rutas en el espacio operacional. Por otro lado, la forma parcial precisa crear inicialmente una parte del prisma, siendo más probable la generación de los nodos que más se aproximen al punto de destino. Cada vez que se llega a un punto nuevo, se calculan todas las probabilidades que tiene éste de escoger a cualquiera de sus vecinos. La probabilidad de cada uno se calcula como se indica en (3.8), siendo  $d_i$  la distancia del punto hasta el destino y  $d_{total}$  la suma de todas las distancias  $d_i$ . La elección del vecino se realiza de forma aleatoria, siendo más probable elegir el que se encuentra más cercano al destino. El cálculo de estas probabilidades sólo se realiza si se trata de un nodo nuevo (no visitado), ya que, si se ha visitado y se decide crear otro nuevo, ya dispondríamos del resultado obtenido anteriormente. Obviamente sólo se contemplan las probabilidades asociadas a generar nuevos nodos, excluyendo la probabilidad de elección de los nodos ya existentes.

$$p_i = \frac{d_i}{d_{total}} \quad (3.8)$$

De esta forma, se les facilita a las hormigas un espacio operativo menos complejo. Esto supone por un lado, acelerar la velocidad de convergencia del algoritmo, y por otro, aumentar la probabilidad de obtener la ruta óptima al no contemplarse los nodos por los que se generan las rutas más largas. En la Figura 17 se aprecia un pseudocódigo que facilita su entendimiento.



---

**Generación parcial del espacio operativo**

---

```
for 0 to (número de caminos base a generar)  
  while (el nodo no coincida con el destino) do  
    if (nodo no visitado)  
      Cálculo de las probabilidades de elección de los vecinos  
      Elección de uno de ellos  
      Eliminación de la probabilidad asociada a escoger ese nodo en el punto actual  
      Introducción al espacio operacional del nodo elegido  
    else  
      Recuperar cálculos de probabilidades de los vecinos  
      Elección de uno de ellos  
      Eliminación de la probabilidad asociada a escoger ese nodo en el punto actual  
      Introducción al espacio operacional del nodo elegido
```

---

**Figura 17. Pseudocódigo de la generación parcial del espacio operativo.**

No obstante, aunque esta metodología está disponible dentro del algoritmo, ningún ejercicio de los realizados se encuentra definido de esta manera. Esto es debido a la posibilidad de no generar un nodo por el que pase la ruta óptima, lo que supondría que no se pudiera encontrar el mejor camino. Actualmente, se está desarrollando la posibilidad de que las hormigas puedan generar nuevos nodos no visitados para solucionar este problema y, de esta manera, poder aplicarlo garantizando así al menos cierta probabilidad de que se generen todos los nodos que definen al óptimo. De alguna manera, con esta metodología, se podría generar un algoritmo que permitiera enfrentarse a problemas dinámicos, al contemplarse un grafo funcionalmente activo. Actualmente se está trabajando en esta mejora y su resolución será objeto de una futura publicación.

Mientras se genera el espacio, se analizan los obstáculos del entorno de forma que los nodos definidos dentro de éstos se vuelven inaccesibles. A continuación, se calculan todas las

configuraciones adyacentes para el espacio operacional, teniendo en cuenta las colisiones con los obstáculos, es decir, las distintas configuraciones viables que puede tomar el robot desde un punto hasta sus vecinos sin que colisione con alguno de los obstáculos existentes. Para el cálculo de estas configuraciones se ha tenido en cuenta que una configuración  $C^j$ , se denomina adyacente a otra,  $C^{j+1}$  cuando es accesible y en ella se cumplen las condiciones siguientes:

- La distancia entre la posición del manipulador  $\varphi_4^{j+1}$  correspondiente a la configuración  $C^{j+1}$  y su inmediata anterior en la configuración  $C^j$ , es inferior al obstáculo de menor tamaño.

$$\|\varphi_4^{j+1} - \varphi_4^j\| < \text{tamaño del menor obstáculo} \quad (3.9)$$

- Entre las dos configuraciones, no existe un obstáculo que provoque la colisión con alguno de los brazos del robot.
- La distancia entre puntos significativos satisface a:

$$\|\varphi_4^{j+1} - \varphi_4^j\| < 2 * \min(\text{pasos del robot}) \quad (3.10)$$

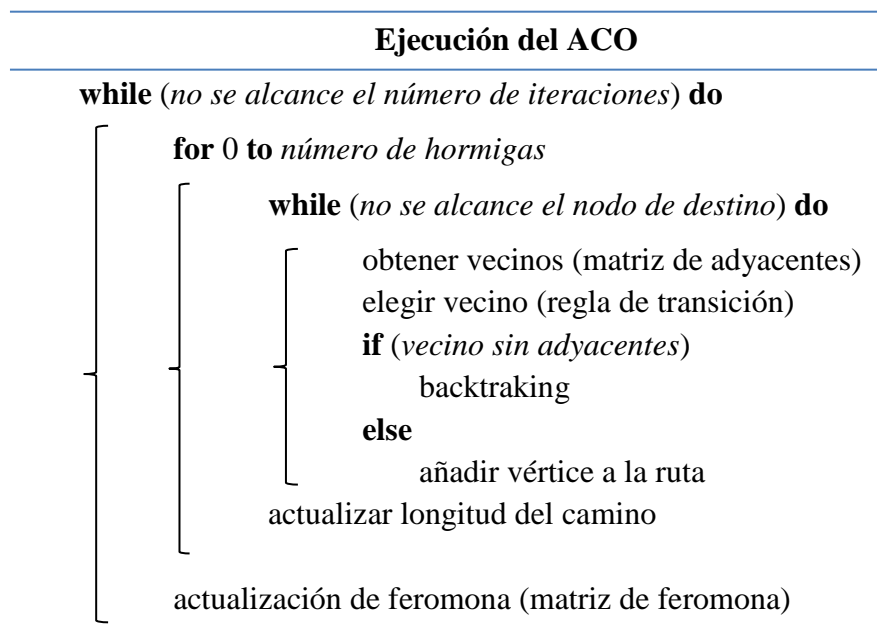
Descrito todo el espacio operativo, se puede aplicar la optimización de rutas, aunque, a priori hay que definir las restricciones del problema. En nuestro caso, dado que se quiere optimizar la ruta entre dos configuraciones previamente definidas, todas las hormigas artificiales empezarán siempre desde la posición inicial. Por otro lado, se restringe la posibilidad de retroceder por el espacio operativo, de forma que sólo se puede avanzar hacia el destino. Con esta restricción se aumenta significativamente la velocidad de convergencia del algoritmo. No obstante, induce nuevos problemas asociados a la existencia de rutas cortadas por obstáculos, dado que sólo se puede avanzar hacia el destino y no retroceder. Como solución, se dota a las hormigas de la posibilidad de hacer “backtracking”. La función del backtracking, incluida en el algoritmo, posiciona a la hormiga en el nodo anterior y borra el nodo que produce un recorrido fallido. La eliminación de este nodo se hace a la vez en el espacio operativo y en la ruta actual de la hormiga. La acción del backtracking puede desencadenar a su vez otra, ya que al borrar el nodo del espacio, el inmediato anterior ha podido quedarse sin vecinos, de forma que se encuentre cortada la ruta de nuevo.

Para todo esto, el algoritmo hace uso de una matriz (denominada de adyacentes) de  $n \times 7$ , donde  $n$  corresponde con el número de nodos generados y 7 con el máximo número de arcos/vecinos que puede tener un mismo punto. Para verificar si el nodo  $i$  tiene por vecino al nodo  $k$ , simplemente se comprueba si en la fila  $i$  de la matriz, alguna columna coincide con el nodo  $k$ . Si esto ocurre, se podrá asegurar que existe un arco entre ellos.

La información de los niveles de feromona de cada arco corresponde también a una matriz de  $n \times 7$ , que a su vez utiliza la matriz de adyacentes. El valor de feromona que se asignará al arco  $ik$  se define en la matriz de feromona, en la misma posición (fila y columna) que en la matriz de adyacentes. De modo, que si el nodo  $i$  tiene en la 3ª columna al nodo  $k$ , el valor de feromona depositado en su arco se encontrará en la fila  $i$  y 3ª columna de la matriz de feromona.

El algoritmo que se ha desarrollado para recorrer el espacio de trabajo está basado en la Colonia Simple de Hormigas, la cual sigue las bases de todo ACO: la construcción de rutas por las hormigas artificiales y la actualización del nivel de feromona de todos los arcos. No se utiliza ninguna variación de las definidas previamente en 3.1.4. No obstante, se han llevado a cabo las modificaciones descritas anteriormente para poder realizar la optimización.

La optimización de rutas puede resumirse en este pseudocódigo:



**Figura 18. Pseudocódigo de ACO.**

# Capítulo 4

## Estimación estadística de los parámetros del ACO

Muchos experimentos tienen como objetivo determinar cómo afecta alguna variable independiente a una dependiente, por ejemplo, temperatura, tiempo... Dado que el ACO es un algoritmo probabilístico, que se basa en las elecciones de las hormigas artificiales, conviene estudiar cómo afectan los parámetros que lo componen con las elecciones que toman las hormigas artificiales. Con ello, no sólo se puede conseguir converger en el óptimo con mayor probabilidad, sino también reducir los tiempos computacionales del algoritmo.

Se pueden hacer numerosos estudios estadísticos como funciones de fiabilidad a partir de la tasa de fallos que se obtiene. No obstante, dado que se quiere estimar cómo influyen los valores de los parámetros del algoritmo, el estudio a realizar es el llamado análisis de varianza (ANOVA). Con él se puede comprobar cómo reacciona el algoritmo frente a un cambio en

cada uno de sus parámetros y, por tanto, llegar a estimar sus valores de modo óptimo para cualquier entorno.

## 4.1. Análisis de varianza (ANOVA)

El análisis de varianza, además de en la descomposición de la variación total, se basa también en la comparación del cambio INTRA-grupos y el cambio de ENTRE-grupos. El cociente entre estos dos grupos indicará las diferencias entre ambos: un valor elevado significa mayor diferencia y por tanto, la posibilidad de que el factor independiente influya en el dependiente en mayor medida.

Por otro lado, el valor que determina si la hipótesis realizada es aceptable o es nula, recae sobre el p-valor. Si éste es mayor que 0.05, se hablará de hipótesis nula, es decir, no existe una diferencia estadísticamente significativa entre las dos variables analizadas. Si por el contrario es menor, podremos decir que existe una relación entre las variables con un nivel de confianza del 95,0%.

## 4.2. Entorno desarrollado para el estudio

El espacio de trabajo formado para el estudio se define con el posicionamiento del nodo inicial en el (0, 0, 0) y del nodo final en el (20, 20, 20). Los pasos entre nodos son de 1 unidad en cada dirección X, Y y Z. Se forma así (como se explicó en el punto 2.3.1) un ortoedro con una diagonal principal contenida en estos puntos. Para el estudio estadístico, no es preciso definir ningún obstáculo. En consecuencia, hay 9261 nodos útiles y la mayoría de ellos disponen de siete arcos por los que avanzar.

Los niveles de feromona iniciales influirán en la velocidad de convergencia del algoritmo, por lo que se sugiere que se encuentren entre 0 y 1. Como los niveles se actualizarán con valores inferiores a 1 (inversa de la longitud recorrida), conviene que el impacto de las primeras iteraciones se refleje cuanto antes porque es un espacio grande y es deseable que, en un principio, se favorezca la exploración. Es por esto, que los niveles iniciales se han fijado a 1.

Para los primeros estudios, se ha decidido ir modificando el número de hormigas e iteraciones para analizar su impacto. Los parámetros prefijados han sido:

- $\alpha = 1$  (peso del valor memorístico).
- $\beta = 1$  (peso del valor heurístico).
- $\rho = 0.1$  (ratio de evaporación).

Cuando se ha calculado el impacto que producen el número de hormigas e iteraciones en los tiempos de ejecución y rutas óptimas, se fijan sus valores a los óptimos en este entorno. Posteriormente, se procede a analizar cómo afecta la evolución de las variables fijadas anteriormente ( $\alpha$ ,  $\beta$  y  $\rho$ ).

## 4.3. Resultados obtenidos

De los análisis se ha obtenido un p-value inferior a 0.05, por lo que podemos asegurar que existe una diferencia estadísticamente significativa entre el tiempo de ejecución/distancia óptima y los parámetros del ACO, con un nivel de confianza del 95,0%. Además, se han realizado 20 “runs o ejecuciones” por iteración para poder comprobar la desviación.

### 4.3.1. Varianza del número de hormigas

Comenzando por el número de hormigas, se fijan los parámetros de la Tabla 2 y se procede a valorar su evolución, en relación con la distancia óptima conseguida y con el tiempo de ejecución.

**Tabla 2. Parámetros fijados para el estudio de la influencia del número de hormigas.**

Iteraciones	$\alpha$	$\beta$	$\rho$	$\tau_0$
250	1	1	0,1	1

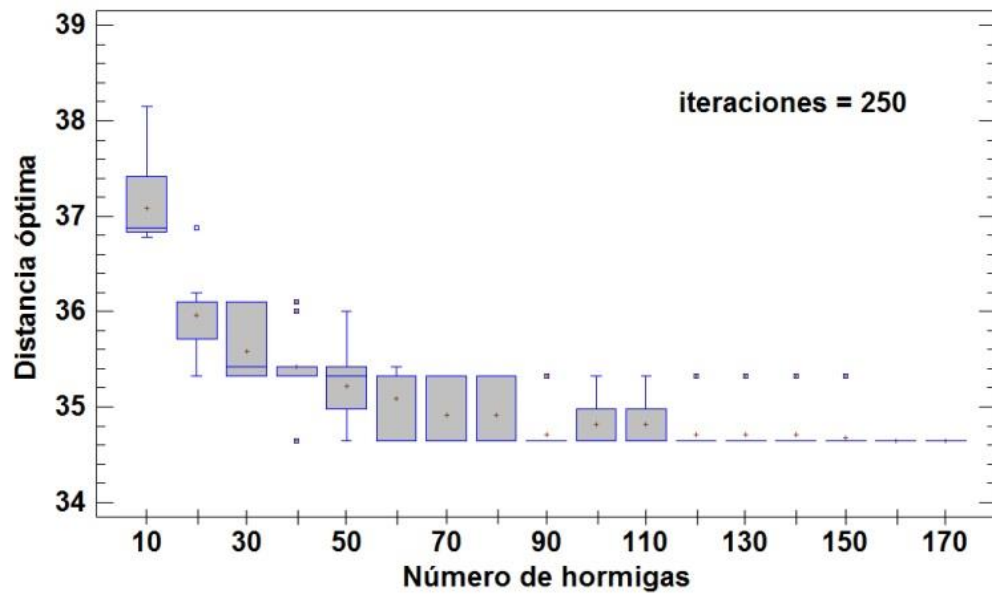


Figura 19. Distancia óptima en función del número de hormigas.

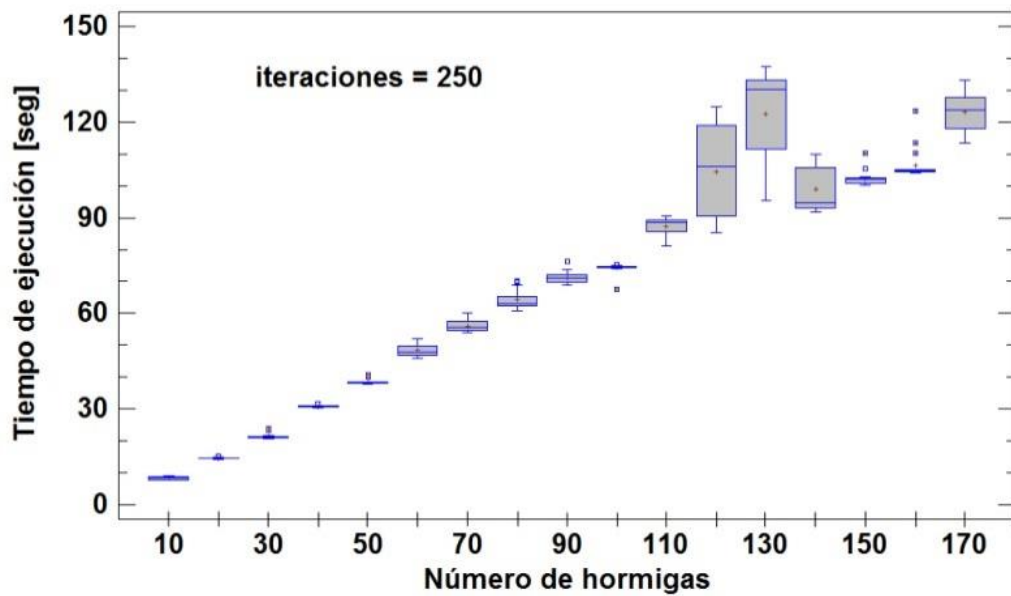


Figura 20. Tiempo de ejecución en función del número de hormigas.

**Tabla 3. Distancia óptima en función del número hormigas.**

Nº de hormigas	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
10	20	37,0853	0,393496	1,06%	36,7839	38,1482	1,3643
20	20	35,9653	0,389899	1,08%	35,3232	36,8803	1,5571
30	20	35,5809	0,352074	0,99%	35,3232	36,1017	0,7785
40	20	35,4251	0,391493	1,11%	34,641	36,1017	1,4607
50	20	35,2109	0,368945	1,05%	34,641	36,0053	1,3643
60	20	35,0892	0,338139	0,96%	34,641	35,4196	0,7786
70	20	34,9139	0,342891	0,98%	34,641	35,3232	0,6822
80	20	34,9139	0,342891	0,98%	34,641	35,3232	0,6822
90	20	34,7092	0,209977	0,60%	34,641	35,3232	0,6822
100	20	34,8115	0,303075	0,87%	34,641	35,3232	0,6822
110	20	34,8115	0,303075	0,87%	34,641	35,3232	0,6822
120	20	34,7092	0,209977	0,60%	34,641	35,3232	0,6822
130	20	34,7092	0,209977	0,60%	34,641	35,3232	0,6822
140	20	34,7092	0,209977	0,60%	34,641	35,3232	0,6822
150	20	34,6751	0,152545	0,44%	34,641	35,3232	0,6822
160	20	34,641	0	0%	34,641	34,641	0
170	20	34,641	0	0%	34,641	34,641	0
Total	340	35,0942	0,681301	1,94%	34,641	38,1482	3,5072
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	129,746	16	8,10915	94,87	0		
Intra grupos	27,6076	323	0,0854723				
Total (Corr.)	157,354	339					

El número de hormigas afecta directamente a la probabilidad de exploración del entorno de modo que, una población pequeña, no es capaz de obtener el mejor camino. A medida que se incrementa este parámetro, se obtienen menores distancias hasta que se llega a un nivel en el cual, únicamente se incrementará el tiempo de ejecución, ya que el camino óptimo siempre es encontrado.

La mejor ruta posible se encuentra en la diagonal principal del modelo: un ortoedro de 20x20x20, cuyo valor es igual a 34.641 unidades.



**Tabla 4. Tiempo de ejecución en función del número hormigas.**

Nº de hormigas	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
10	20	8,21652	0,444829	5,41%	7,66344	8,82751	1,16407
20	20	14,5236	0,201481	1,39%	14,2828	15,1729	0,8901
30	20	21,3603	0,797751	3,73%	20,5902	23,9304	3,3402
40	20	30,7196	0,324607	1,06%	30,3107	31,6558	1,3451
50	20	38,3675	0,724275	1,89%	37,8122	40,8063	2,9941
60	20	48,3173	1,84186	3,81%	45,8046	52,036	6,2314
70	20	56,0487	1,89335	3,38%	53,8651	59,9794	6,1143
80	20	64,2814	2,99925	4,67%	60,6885	70,21	9,5215
90	20	71,2195	1,8854	2,65%	68,9119	76,4054	7,4935
100	20	74,2778	1,6096	2,17%	67,5639	75,4133	7,8494
110	20	87,2429	2,88787	3,31%	81,1096	90,6082	9,4986
120	20	104,556	14,1233	13,51%	85,2679	124,961	39,6931
130	20	122,67	15,8558	12,93%	95,3195	137,447	42,1275
140	20	98,9753	6,73794	6,81%	91,8533	109,843	17,9897
150	20	102,283	2,20695	2,16%	100,194	110,182	9,988
160	20	106,365	4,6176	4,34%	104,204	123,418	19,214
170	20	123,286	6,01315	4,88%	113,491	133,188	19,697
Total	340	68,983	37,1349	53,83%	7,66344	137,447	129,784
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	456261	16	28516,3	820,84	0		
Intra grupos	11221,2	323	34,7405				
Total (Corr.)	467482	339					

### 4.3.2. Varianza del número de iteraciones

Para estudiar la influencia de las iteraciones se fijan los parámetros según la Tabla 5:

**Tabla 5. Parámetros fijados para el estudio de la influencia del número de iteraciones.**

Hormigas	$\alpha$	$\beta$	$\rho$	$\tau_0$
200	1	1	0,1	1

Se ha optado por sobrecargar el número de hormigas a 200, aunque el estudio anterior muestre que con una colonia de 160 puede encontrarse el óptimo. Con esto, se quiere

determinar cómo afecta este parámetro, suponiendo los demás óptimos y sin tener en cuenta las iteraciones definidas anteriormente.

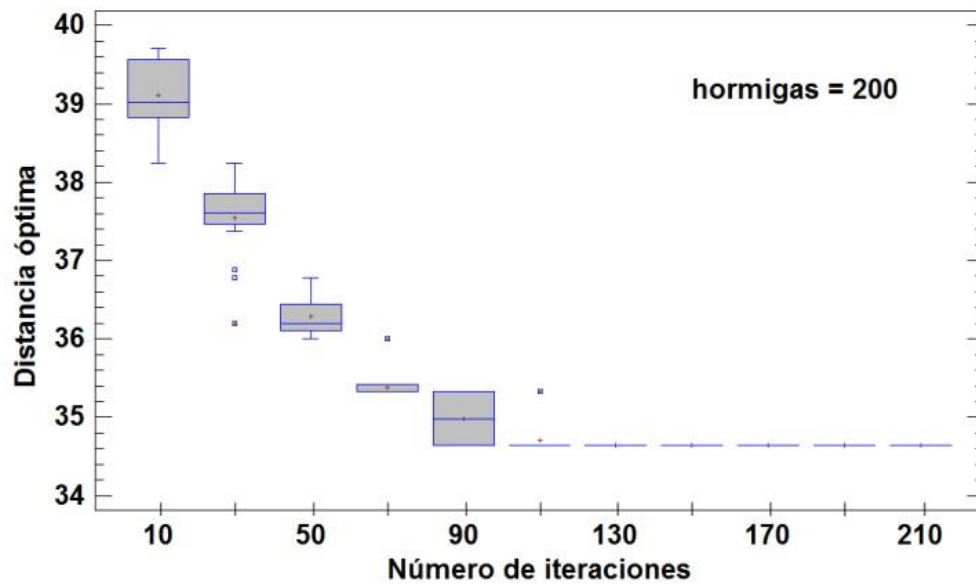


Figura 21. Distancia óptima en función del número iteraciones.

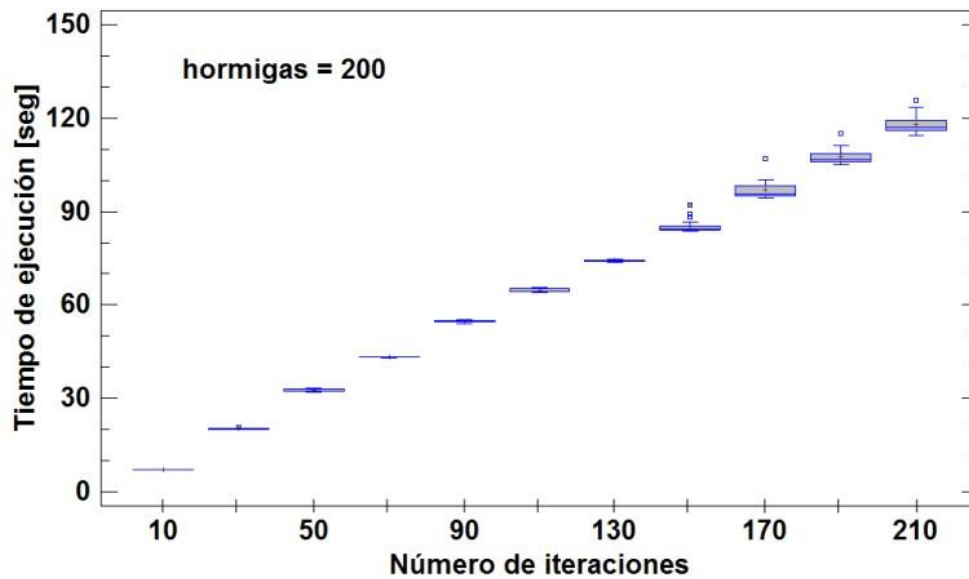


Figura 22. Tiempo de ejecución en función del número de iteraciones.

**Tabla 6. Distancia óptima en función del número iteraciones.**

Nºde iteraciones	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
10	20	39,1169	0,406248	1,04%	38,2446	39,7053	1,4607
30	20	37,5431	0,518339	1,38%	36,1981	38,2446	2,0465
50	20	36,2819	0,293707	0,81%	36,0053	36,7839	0,7786
70	20	35,3814	0,152862	0,43%	35,3232	36,0053	0,6821
90	20	34,9821	0,349961	1,00%	34,641	35,3232	0,6822
110	20	34,7092	0,209977	0,60%	34,641	35,3232	0,6822
130	20	34,641	0	0%	34,641	34,641	0
150	20	34,641	0	0%	34,641	34,641	0
170	20	34,641	0	0%	34,641	34,641	0
190	20	34,641	0	0%	34,641	34,641	0
210	20	34,641	0	0%	34,641	34,641	0
Total	220	35,5654	1,4522	4,08%	34,641	39,7053	5,0643
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	448,359	10	44,8359	694,73	0		
Intra grupos	13,4882	209	0,064537				
Total (Corr.)	461,847	219					

**Tabla 7. Tiempo de ejecución en función del número iteraciones.**

Nº de iteraciones	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
10	20	6,918	0,0144554	0,21%	6,88939	6,9444	0,05501
30	20	20,1314	0,185779	0,92%	19,9501	20,6282	0,6781
50	20	32,4351	0,421651	1,30%	31,9288	33,1589	1,2301
70	20	43,2413	0,106802	0,25%	43,0785	43,4265	0,348
90	20	54,6254	0,323466	0,59%	53,9121	55,2932	1,3811
110	20	64,6947	0,598193	0,92%	64,0707	65,7128	1,6421
130	20	74,2119	0,224448	0,30%	73,7212	74,6323	0,9111
150	20	85,364	2,1667	2,54%	83,8768	92,2343	8,3575
170	20	97,0218	2,96193	3,05%	94,3614	107,05	12,6886
190	20	107,628	2,61977	2,43%	105,026	115,267	10,241
210	20	117,975	2,89839	2,46%	114,492	125,949	11,457
Total	220	64,0224	34,7125	54,22%	6,88939	125,949	119,06
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	263325	10	26332,5	9829,5	0		
Intra grupos	559,896	209	2,67893				
Total (Corr.)	263885	219					

Dado que el número de iteraciones son las veces que salen todas las hormigas y por lo tanto, se actualizan los niveles de feromona, cuantas más se usen, más se favorecerá al aprendizaje de la colonia, es decir, la deposición de feromona. No obstante, se alcanza a un punto en el que su incremento no supone una mejora en el cálculo del mejor camino, ya que siempre se alcanza el óptimo. Con esto, sólo se estarían incrementando los valores memorísticos de los arcos en la/s rutas óptimas, lo que supone un incremento en el tiempo de ejecución, al tener que realizarse más salidas en la colonia.

### 4.3.3. Estudios de $\alpha$ , $\beta$ y $\rho$

En los siguientes estudios, se analizan los efectos de  $\alpha$ ,  $\beta$  y  $\rho$ , con los valores indicados en la Tabla 8. Es obvio indicar que, cuando se analiza  $\alpha$ ,  $\beta$  o  $\rho$ , su valor sí se modificará y no lo hará el de los otros parámetros.

Tabla 8. Parámetros fijados para el estudio de la influencia de  $\alpha$ ,  $\beta$  o  $\rho$ .

Hormigas	Iteraciones	$\alpha$	$\beta$	$\rho$	$\tau_0$
130	150	1	1	0,1	1

#### 4.3.3.1. Varianza de $\alpha$

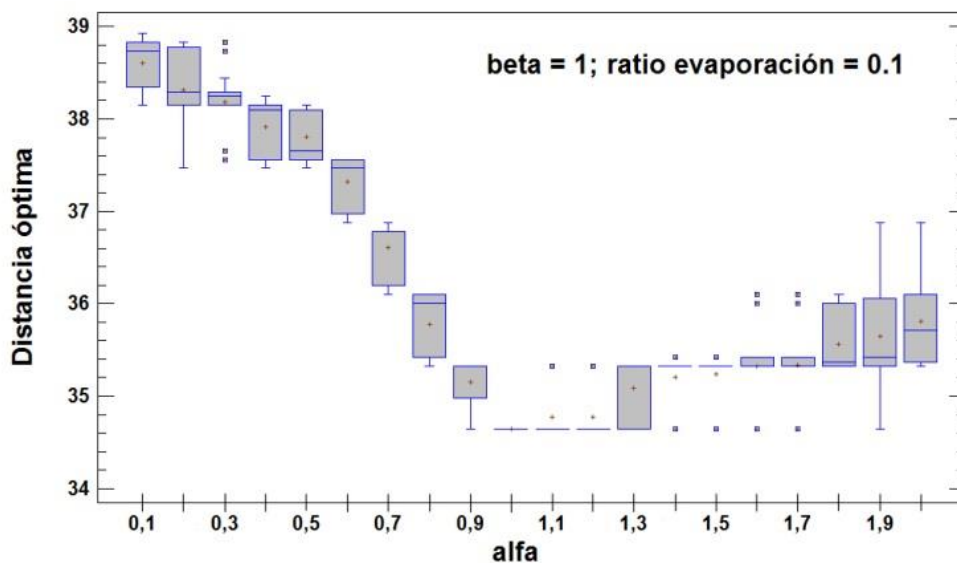
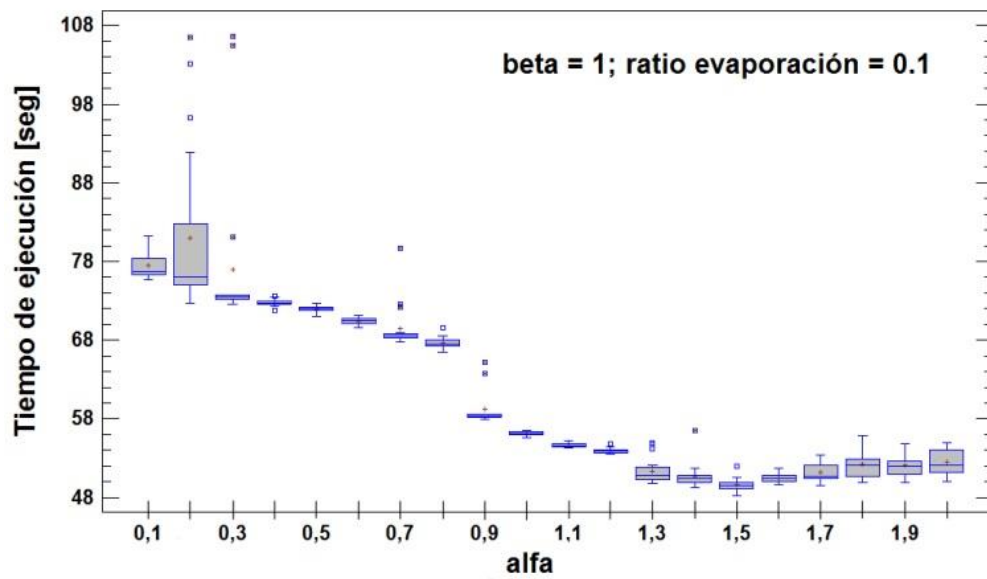


Figura 23. Distancia óptima en función de  $\alpha$ .



**Figura 24.** Tiempo de ejecución en función de  $\alpha$ .

El parámetro  $\alpha$  se corresponde con el peso que la simulación asigna a los valores memorísticos, de modo que favorece la elección de caminos que dispongan de un nivel de feromona mayor a otros. En la Figura 23 vemos su evolución afectando al camino óptimo, si  $\alpha$  toma valores pequeños, se valorará mucho más el factor heurístico del entorno. De esta forma, la colonia será propensa a elegir los caminos que considere más cortos, sin tener en cuenta un aprendizaje conjunto. Por otro lado, si se potencia más el valor memorístico, las hormigas tenderán a ir por los caminos con mayor concentración de feromona, sin tener en cuenta la distancia entre los puntos, lo que recargaría siempre los mismos caminos, haciendo que se evaporase el valor memorístico de los demás. Este valor resulta óptimo cuando se le da el mismo peso que al parámetro heurístico  $\beta$ , es decir, cuando los dos influyen por igual en la elección del camino.

**Tabla 9. Distancia óptima en función de  $\alpha$ .**

alfa	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
0,1	20	38,6053	0,270958	0,70%	38,1482	38,9267	0,7785
0,2	20	38,3084	0,431709	1,13%	37,466	38,8304	1,3644
0,3	20	38,1812	0,333867	0,87%	37,5624	38,8304	1,268
0,4	20	37,9187	0,288993	0,76%	37,466	38,2446	0,7786
0,5	20	37,8015	0,269846	0,71%	37,466	38,1482	0,6822
0,6	20	37,3192	0,273271	0,73%	36,8803	37,5624	0,6821
0,7	20	36,613	0,304841	0,83%	36,1017	36,8803	0,7786
0,8	20	35,7755	0,342404	0,96%	35,3232	36,1017	0,7785
0,9	20	35,1527	0,303075	0,86%	34,641	35,3232	0,6822
1	20	34,641	0	0%	34,641	34,641	0
1,1	20	34,7774	0,279969	0,81%	34,641	35,3232	0,6822
1,2	20	34,7774	0,279969	0,81%	34,641	35,3232	0,6822
1,3	20	35,0844	0,333842	0,95%	34,641	35,3232	0,6822
1,4	20	35,206	0,29238	0,83%	34,641	35,4196	0,7786
1,5	20	35,2402	0,261113	0,74%	34,641	35,4196	0,7786
1,6	20	35,3228	0,426363	1,21%	34,641	36,1017	1,4607
1,7	20	35,3325	0,427403	1,21%	34,641	36,1017	1,4607
1,8	20	35,5664	0,340983	0,96%	35,3232	36,1017	0,7785
1,9	20	35,6442	0,496611	1,39%	34,641	36,8803	2,2393
2	20	35,8048	0,496978	1,39%	35,3232	36,8803	1,5571
Total	400	36,1536	1,35332	3,74%	34,641	38,9267	4,2857
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	687,047	19	36,1604	314,37	0		
Intra grupos	43,7097	380	0,115026				
Total (Corr.)	730,757	399					

Tabla 10. Tiempo de ejecución en función de  $\alpha$ .

alfa	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
0,1	20	77,4877	1,7955	2,32%	75,6763	81,2676	5,5913
0,2	20	81,0361	10,1888	12,57%	72,6952	106,434	33,7388
0,3	20	76,9661	10,0982	13,12%	72,5161	106,569	34,0529
0,4	20	72,7055	0,430896	0,59%	71,8201	73,6212	1,8011
0,5	20	71,953	0,34805	0,48%	71,0611	72,6392	1,5781
0,6	20	70,4073	0,475024	0,67%	69,549	71,0811	1,5321
0,7	20	69,3929	2,72705	3,93%	67,8129	79,7016	11,8887
0,8	20	67,6537	0,740717	1,09%	66,4648	69,634	3,1692
0,9	20	59,2599	2,36754	4,00%	57,9683	65,1737	7,2054
1	20	56,1468	0,22961	0,41%	55,5922	56,4772	0,885
1,1	20	54,6002	0,271389	0,50%	54,2471	55,2342	0,9871
1,2	20	53,89	0,312313	0,58%	53,5421	54,7831	1,241
1,3	20	51,3313	1,55415	3,03%	49,8188	54,9691	5,1503
1,4	20	50,6487	1,49814	2,96%	49,2216	56,4652	7,2436
1,5	20	49,6088	0,796593	1,61%	48,2398	51,937	3,6972
1,6	20	50,4691	0,610642	1,21%	49,6128	51,749	2,1362
1,7	20	51,1303	1,18704	2,32%	49,5208	53,4351	3,9143
1,8	20	52,1777	1,57883	3,03%	49,9489	55,7952	5,8463
1,9	20	52,087	1,48284	2,85%	49,8539	54,8241	4,9702
2	20	52,5014	1,55449	2,96%	50,0259	54,8911	4,8652
Total	400	61,0727	11,1949	18,33%	48,2398	106,569	58,3292
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	45491,8	19	2394,31	201,59	0		
Intra grupos	4513,27	380	11,877				
Total (Corr.)	50005,1	399					

### 4.3.3.2. Varianza de $\beta$

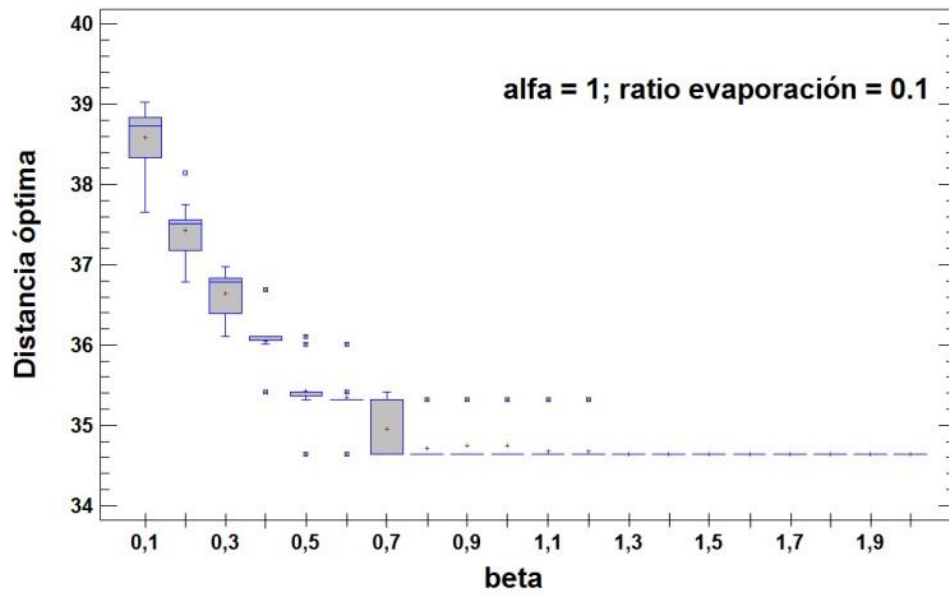


Figura 25. Distancia óptima en función de  $\beta$ .

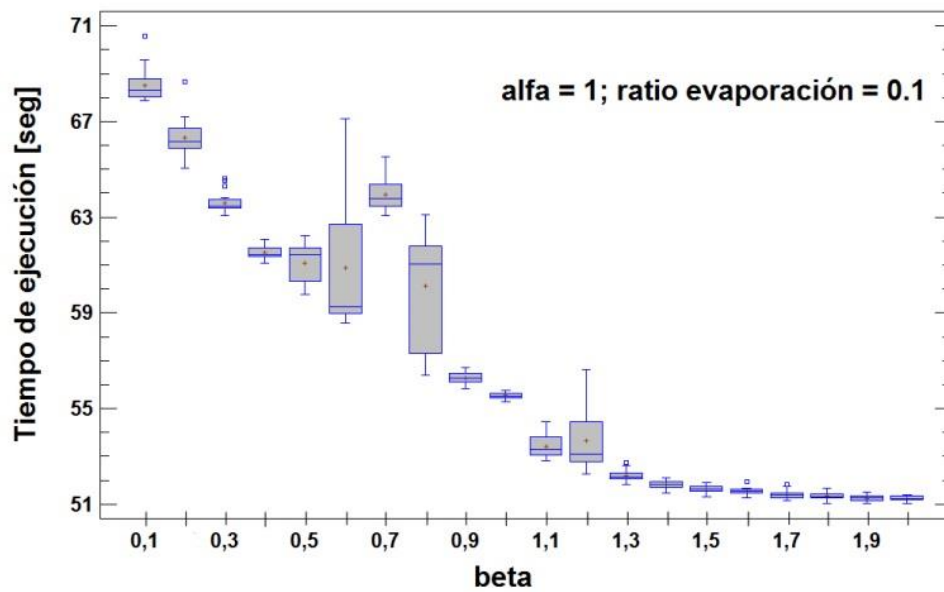


Figura 26. Tiempo de ejecución en función de  $\beta$ .



**Tabla 11. Distancia óptima en función de  $\beta$ .**

beta	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
0,1	20	38,5808	0,354668	0,92%	37,6588	39,0231	1,3643
0,2	20	37,4264	0,387729	1,04%	36,7839	38,1482	1,3643
0,3	20	36,6423	0,297343	0,81%	36,1017	36,9766	0,8749
0,4	20	36,0483	0,255937	0,71%	35,4196	36,6875	1,2679
0,5	20	35,4248	0,276363	0,78%	34,641	36,1017	1,4607
0,6	20	35,3377	0,224119	0,63%	34,641	36,0053	1,3643
0,7	20	34,9528	0,354288	1,01%	34,641	35,4196	0,7786
0,8	20	34,7092	0,209977	0,60%	34,641	35,3232	0,6822
0,9	20	34,7433	0,249922	0,72%	34,641	35,3232	0,6822
1	20	34,7433	0,249922	0,72%	34,641	35,3232	0,6822
1,1	20	34,6751	0,152545	0,44%	34,641	35,3232	0,6822
1,2	20	34,6751	0,152545	0,44%	34,641	35,3232	0,6822
1,3	20	34,641	0	0%	34,641	34,641	0
1,4	20	34,641	0	0%	34,641	34,641	0
1,5	20	34,641	0	0%	34,641	34,641	0
1,6	20	34,641	0	0%	34,641	34,641	0
1,7	20	34,641	0	0%	34,641	34,641	0
1,8	20	34,641	0	0%	34,641	34,641	0
1,9	20	34,641	0	0%	34,641	34,641	0
2	20	34,641	0	0%	34,641	34,641	0
Total	400	35,2544	1,09269	3,10%	34,641	39,0231	4,3821
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	459,338	19	24,1757	538,6	0		
Intra grupos	17,0566	380	0,0448859				
Total (Corr.)	476,395	399					

El parámetro  $\beta$  corresponde con el peso que se da a los valores heurísticos, de modo que la influencia entre la distancia de dos nodos recae sobre él. En la Figura 25, puede verse la evolución que sigue en la ruta óptima conseguida. Ésta va mejorando a medida que se incrementa  $\beta$ , llegando a valores en los que la óptima siempre es encontrada, por mucho que se aumente. Esto es así porque estamos en un caso ideal, sin obstáculos. No obstante, si estuvieran definidos con unos valores elevados de  $\beta$ , nos encontraríamos peores resultados, al favorecer más la elección por heurísticos, sin tener en cuenta el aprendizaje con los niveles de feromona.

**Tabla 12. Tiempo de ejecución en función de  $\beta$ .**

beta	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
0,1	20	68,5207	0,706764	1,03%	67,8629	70,572	2,7091
0,2	20	66,3375	0,794365	1,20%	65,0537	68,6839	3,6302
0,3	20	63,5978	0,427675	0,67%	63,0736	64,6087	1,5351
0,4	20	61,5067	0,249932	0,41%	61,0925	62,0665	0,974
0,5	20	61,0893	0,794046	1,30%	59,7484	62,2446	2,4962
0,6	20	60,873	2,80791	4,61%	58,5844	67,1348	8,5504
0,7	20	63,9379	0,623638	0,98%	63,0556	65,5177	2,4621
0,8	20	60,1401	2,25615	3,75%	56,3892	63,1036	6,7144
0,9	20	56,2763	0,264904	0,47%	55,8342	56,7132	0,879
1	20	55,5468	0,137202	0,25%	55,2852	55,7702	0,485
1,1	20	53,4226	0,526811	0,99%	52,819	54,4241	1,6051
1,2	20	53,6431	1,20168	2,24%	52,262	56,6102	4,3482
1,3	20	52,1935	0,217994	0,42%	51,808	52,727	0,919
1,4	20	51,821	0,163191	0,31%	51,4499	52,081	0,6311
1,5	20	51,6354	0,139325	0,27%	51,2959	51,92	0,6241
1,6	20	51,5508	0,140216	0,27%	51,2819	51,953	0,6711
1,7	20	51,3943	0,175092	0,34%	51,1439	51,834	0,6901
1,8	20	51,34	0,145518	0,28%	51,0319	51,649	0,6171
1,9	20	51,2403	0,124478	0,24%	51,0299	51,4999	0,47
2	20	51,2394	0,11108	0,22%	51,0179	51,3839	0,366
Total	400	56,8653	5,72789	10,07%	51,0179	70,572	19,5541
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	12760,6	19	671,61	773,15	0		
Intra grupos	330,094	380	0,868668				
Total (Corr.)	13090,7	399					

### 4.3.3.3. Varianza de $\rho$

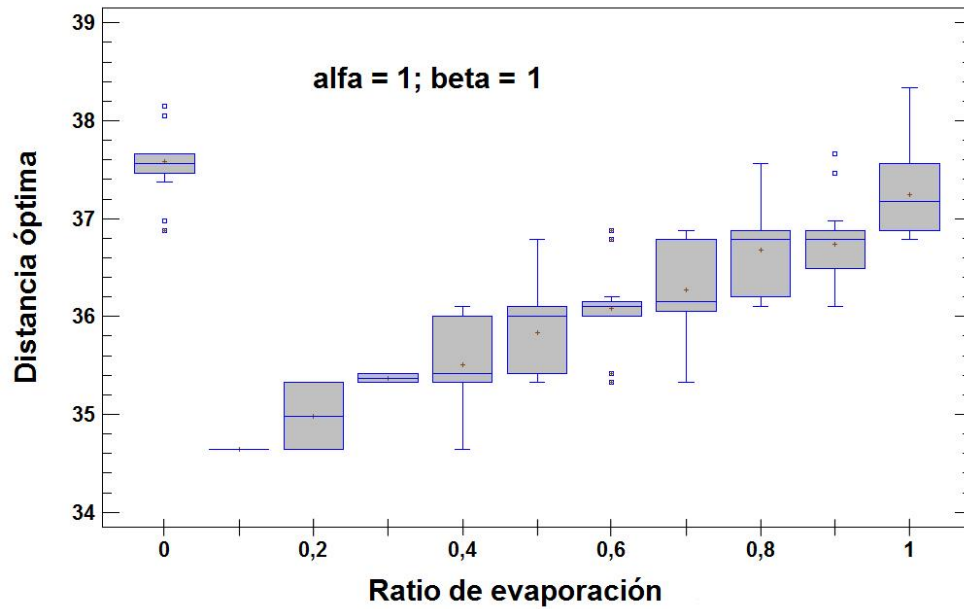


Figura 27. Distancia óptima en función de  $\rho$ .

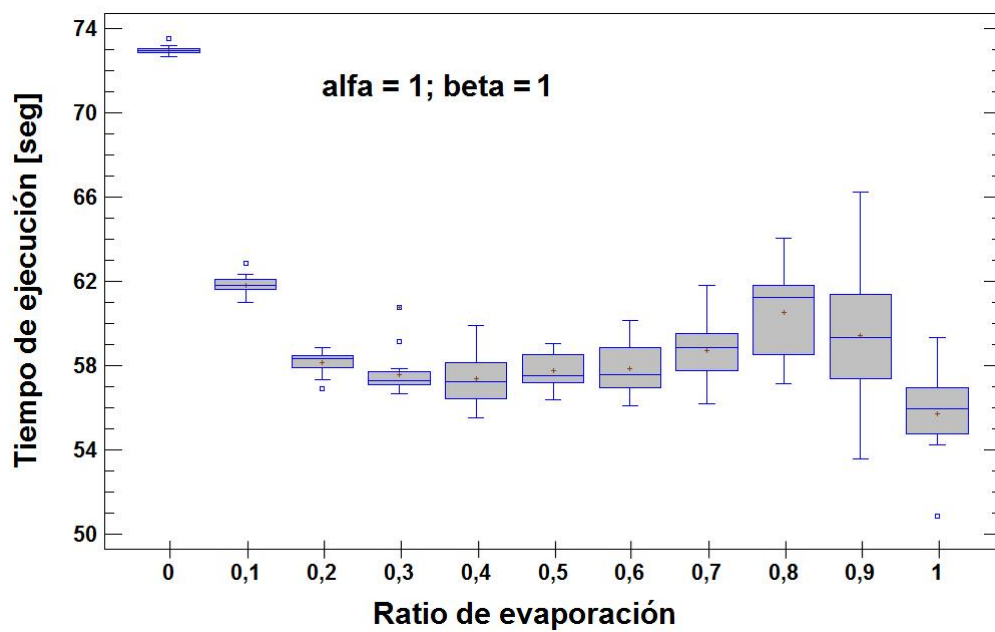


Figura 28. Tiempo de ejecución en función de  $\rho$ .

Tabla 13. Distancia óptima en función de  $\rho$ .

Ratio de evaporación	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
0	20	37,5873	0,331777	0,88%	36,8803	38,1482	1,2679
0,1	20	34,641	0	0%	34,641	34,641	0
0,2	20	34,9821	0,349961	1,00%	34,641	35,3232	0,6822
0,3	20	35,3714	0,0494522	0,14%	35,3232	35,4196	0,0964
0,4	20	35,5078	0,430923	1,21%	34,641	36,1017	1,4607
0,5	20	35,8341	0,403414	1,13%	35,3232	36,7839	1,4607
0,6	20	36,0873	0,468932	1,30%	35,3232	36,8803	1,5571
0,7	20	36,2723	0,455601	1,26%	35,3232	36,8803	1,5571
0,8	20	36,6812	0,412838	1,13%	36,1017	37,5624	1,4607
0,9	20	36,7346	0,411457	1,12%	36,1017	37,6588	1,5571
1	20	37,2458	0,416735	1,12%	36,7839	38,341	1,5571
Total	220	36,0859	0,959055	2,66%	34,641	38,341	3,7
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	172,472	10	17,2472	124,46	0		
Intra grupos	28,9617	209	0,138573				
Total (Corr.)	201,433	219					

Tabla 14. Tiempo de ejecución en función de  $\rho$ .

Ratio de evaporación	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación	Mínimo	Máximo	Rango
0	20	72,9862	0,188416	0,26%	72,6982	73,5252	0,827
0,1	20	61,8212	0,420896	0,68%	60,9975	62,8506	1,8531
0,2	20	58,1462	0,504925	0,87%	56,9133	58,8414	1,9281
0,3	20	57,5571	0,921227	1,60%	56,6642	60,7555	4,0913
0,4	20	57,3881	1,1314	1,97%	55,5002	59,9164	4,4162
0,5	20	57,7767	0,844543	1,46%	56,3562	59,0304	2,6742
0,6	20	57,8441	1,17266	2,03%	56,0682	60,1354	4,0672
0,7	20	58,7145	1,40657	2,40%	56,1782	61,7935	5,6153
0,8	20	60,5022	2,02561	3,35%	57,1143	64,0587	6,9444
0,9	20	59,4388	3,23587	5,44%	53,5481	66,2488	12,7007
1	20	55,7286	1,73237	3,11%	50,8579	59,3384	8,4805
Total	220	59,8094	4,69009	7,84%	50,8579	73,5252	22,6673
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P		
Entre grupos	4356,81	10	435,681	197,72	0		
Intra grupos	460,527	209	2,20348				
Total (Corr.)	4817,34	219					

El ratio de evaporación influye en la velocidad de disminución de los valores memorísticos del entorno. De este modo, si existen caminos que no se transitan, la feromona depositada en ellos tenderá a 0. El hecho de que  $\rho$  tome un valor de 0 implica que sólo se incrementan los niveles de feromona y esto afectará en gran medida en las primeras iteraciones, ya que los caminos no transitados seguirán teniendo un gran valor memorístico en ellos ( $\tau$ ). Por otro lado, valores elevados de  $\rho$  provocan que el incremento de feromona tenga un menor impacto, al reducirse en gran medida todos los niveles al mismo tiempo, con lo que se obtienen unos incrementos simbólicos.

## 4.4. Discusión de los resultados

La elección de los valores de los parámetros del ACO no resulta trivial. En gran medida, estos dependen de las condiciones del entorno que se analiza, como puede ser: del número de posibles nodos adyacentes a uno dado, de las distancias entre cada uno de ellos, del tamaño total del espacio de trabajo. Las restricciones que se definen para cada estudio (como en casos donde se impone la condición de “no volver por nodos visitados”), influyen, en gran medida, a que se prioricen unas características del algoritmo frente a otras.

La exploración del entorno resulta vital para comprobar todas las posibilidades que pueden contener al camino óptimo. Es posible que la ruta definida por nodos adyacentes y muy próximos a uno dado, no establezcan el camino óptimo ya que pueden existir alternativas con menos nodos, aunque más distantes, siendo el resultado final más favorable en este último caso. También puede ocurrir que, por las condiciones de problemas, el nodo más cercano a otro, influya más en un nodo posterior que en ese mismo y, por tanto, su elección en un momento ulterior condiciona una ruta más corta. Este sería el caso de los problemas del viajante, TSP.

Otro aspecto importante a tener en cuenta es el aprendizaje de la colonia. En espacios grandes, en los que existan multitud de posibilidades, se requiere un modo de marcar los recorridos realizados de forma que, en posteriores iteraciones se aprenda de lo analizado y se actúe potenciando los mejores caminos encontrados hasta el momento e incluso se penalice a

los peores elegidos (caso del MMAS, modificación del ACO). De la misma manera que es importante favorecer el aprendizaje, conviene no sobrepasarse a la hora de elegir su impacto, ya que unos valores elevados pueden provocar caer en un mínimo local, es decir, un estancamiento en unas rutas que no corresponden con las óptimas y que imposibiliten la elección de otras por los valores de feromona depositados en ellos.

Podemos concluir que, en este estudio, los valores con los que se asegura encontrar la ruta óptima son los mostrados en la siguiente tabla:

**Tabla 15. Valores óptimos de estadísticos.**

Hormigas	Iteraciones	$\alpha$	$\beta$	$\rho$	$\tau_0$
160	130	1	1	0,1	1

Numerosos autores deciden utilizar el ACO sin antes realizar un estudio de influencia de los parámetros en sus sistemas. Como se ha mencionado anteriormente, éstos pueden variar a los calculados en nuestra simulación por tamaños del entorno, condiciones del problema, etc. Parece por tanto sensato elegirlos de manera cuidadosa y racional, así como analizar la influencia y las consecuencias que tienen. Por ejemplo, en (Deng et al. 2013) se decide empezar con un nivel de feromona inicial bastante elevado, por ello, se ven forzados a incrementar el ratio de evaporación y el peso del valor memorístico  $\alpha$ .

# Capítulo 5

## Experimentos de planificación de caminos

Todos los experimentos se han realizado con un Intel® Core™ 2 Quad CPU Q8200 2.33GHz. El algoritmo se ha desarrollado en C++ utilizando el software Code::Blocks 13.12, se ha hecho uso de la librería *libaco* de (Hammerl 2009). Para la visualización de los resultados se ha utilizado Matlab R2014a.

### 5.1. Ejemplo 1. Obstáculos sencillos

En este apartado se hace uso de obstáculos básicos, tales como esferas, cilindros y planos, con el fin de generar nuevas rutas optimizadas que los eviten. Se empieza sin

establecer ningún impedimento y se van añadiendo hasta definir 3 obstáculos del mismo tipo. Este ejemplo es el mismo propuesto por (Abu-Dakka, et al. 2012) en su publicación, en la que hace uso de algoritmos genéticos.

Los parámetros del ACO usados en este ejemplo son:

**Tabla 16. Parámetros del ACO para el ejemplo 1.**

Hormigas	Iteraciones	$\alpha$	$\beta$	$\rho$	$\tau_0$
5	10	1	1	0,1	1

Las configuraciones inicial y final se detallan en la Tabla 17. A su vez, los obstáculos se definen en la Tabla 18.

**Tabla 17. Configuración inicial y final del robot.**

Articulación	1	2	3	4	5	6
$C^{inicial}(\mathbf{q})$	59.09°	-145.38°	13.03°	1.13°	31.68°	0.00°
$C^{final}(\mathbf{q})$	-34.65°	-169.14°	58.56°	0.00°	15.78°	0.00°

**Tabla 18. Ubicación de los obstáculos.**

	1ª Esfera	2ª Esfera	3ª Esfera
Centro	(-0.85,-0.40,0.50)	(-0.75,-0.70,0.50)	(-0.60,-0.20,0.30)
Radio	0.15	0.15	0.15
	1º Cilindro	2º Cilindro	3º Cilindro
Centro 1	(-0.85,-0.50,0.00)	(-0.75,0.00,0.00)	(-0.70,0.20,0.00)
Centro 2	(-0.85,-0.50,2.00)	(-0.75,0.00,2.00)	(-0.70,0.20,2.00)
Radio	0.15	0.15	0.15
	1º Plano	2º Plano	3º Plano
Punto 1	(-0.70,-0.35,0.00)	(-0.50,0.00,0.00)	(-0.50,0.30,0.00)
Punto 2	(-0.70,-0.35,2.00)	(-0.50,0.00,2.00)	(-0.50,0.30,2.00)
Punto 3	(-1.50,-0.35,2.00)	(-1.30,0.00,2.00)	(-1.30,0.30,2.00)
Punto 4	(-1.50,-0.35,0.00)	(-1.30,0.00,0.00)	(-1.30,0.30,0.00)

A continuación, se muestran las figuras compuestas por el máximo número de obstáculos de cada tipo en el espacio operacional y la ruta óptima generada por el algoritmo para esquivar a cada uno de ellos.



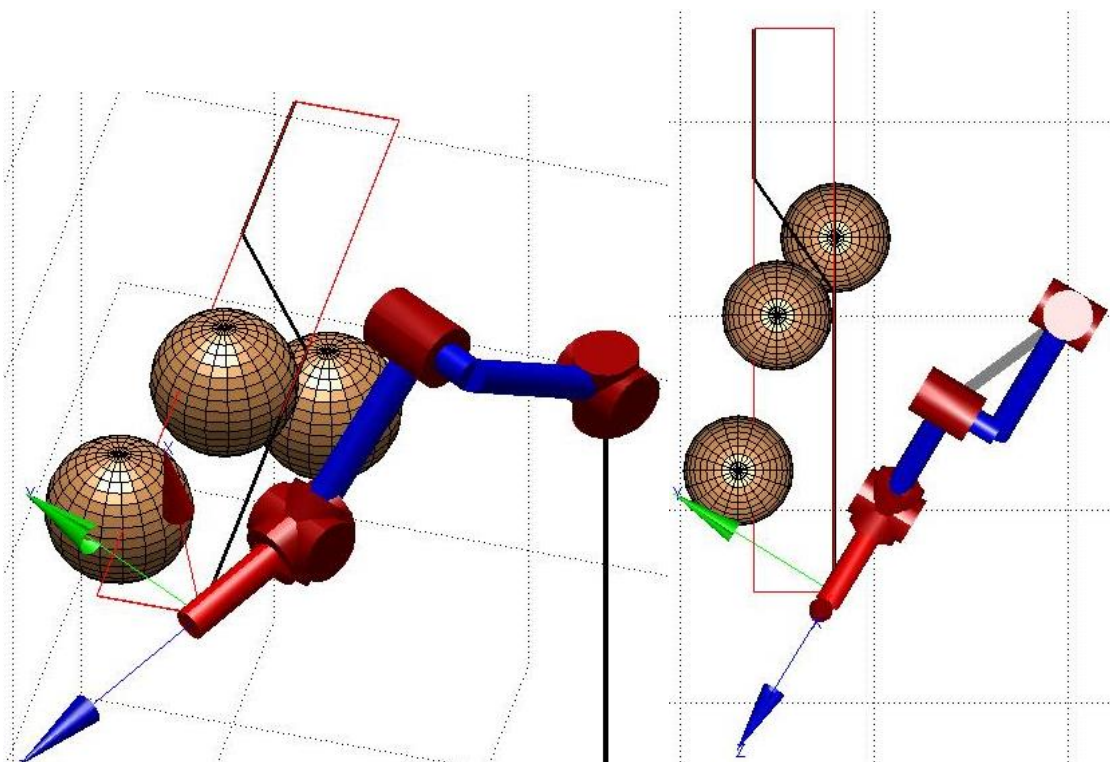


Figura 29. Entorno con esferas.

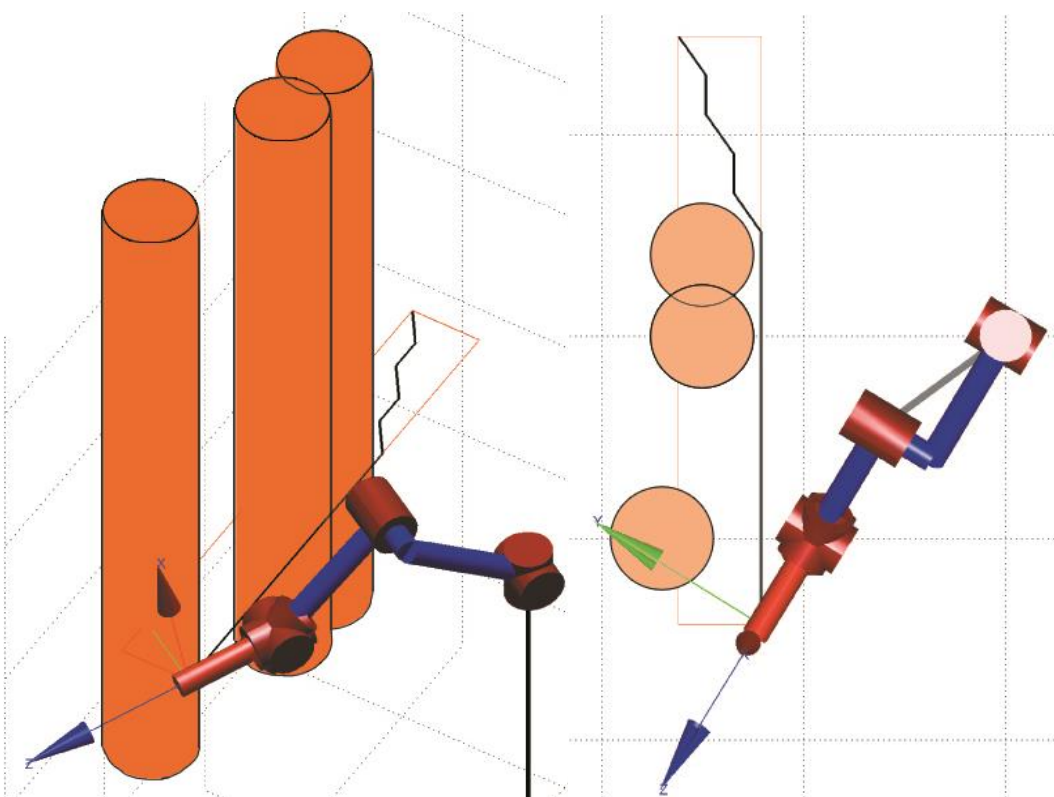


Figura 30. Entorno con cilindros.

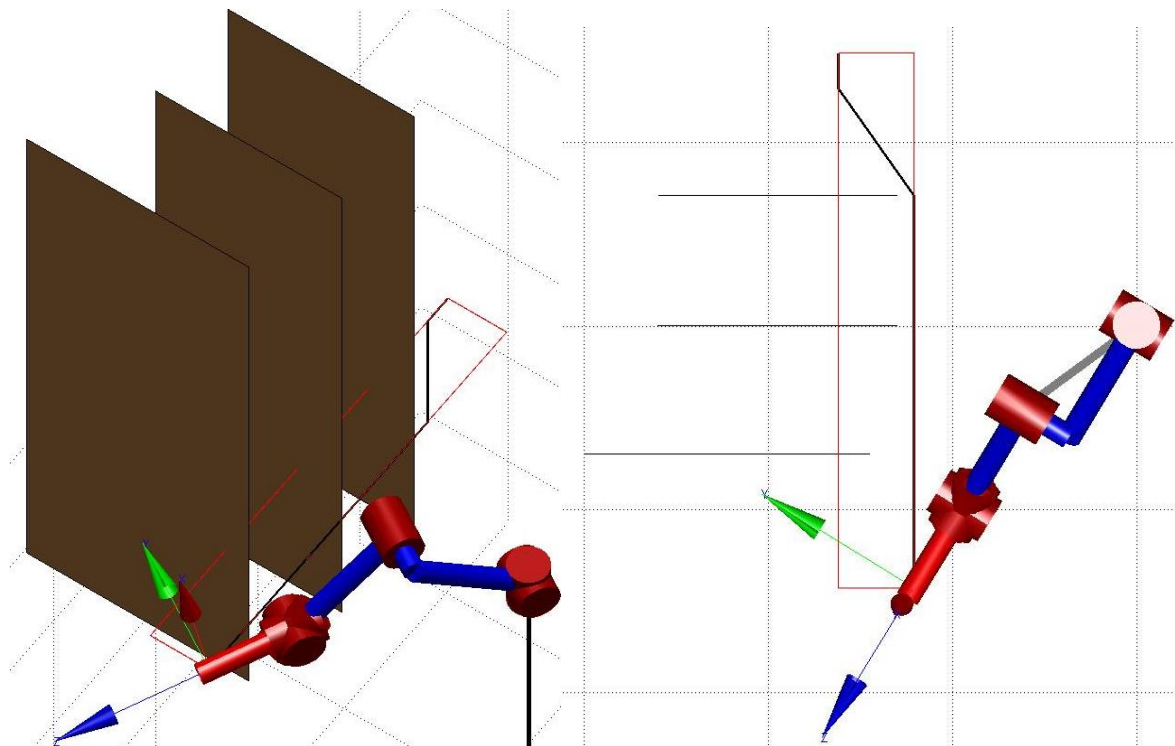


Figura 31. Entorno con planos.

Tabla 19. Resultados ejemplo de obstáculos sencillos comparado con el artículo de (Abu-Dakka, et al. 2012), donde se usan algoritmos genéticos.

	Sin obstáculos					
	ACO	Genético				
Distancia [m]	1.5199	1.5199				
Tiempo ejecución [s]	0.6420	5532				
	1 Esfera		2 Esfera		3 Esfera	
	ACO	Genético	ACO	Genético	ACO	Genético
Distancia [m]	1.5199	1.5199	1.5199	1.5346	1.5199	1.5199
Tiempo ejecución [s]	0.6680	10567	0.6990	3806	0.7380	4013
	1 Cilindro		2 Cilindro		3 Cilindro	
	ACO	Genético	ACO	Genético	ACO	Genético
Distancia [m]	1.5199	1.5199	1.5199	1.6149	1.5199	1.5917
Tiempo ejecución [s]	0.7410	3932	0.7580	8139	0.9381	8366
	1 Plano		2 Plano		3 Plano	
	ACO	Genético	ACO	Genético	ACO	Genético
Distancia [m]	1.5199	1.5364	1.5199	2.1387	1.5199	1.8353
Tiempo ejecución [s]	0.9091	10272	1.0331	19523	1.1561	24322

Los resultado de la Tabla 19 muestran que el algoritmo ha sido capaz de conseguir en todos los casos, un camino óptimo del mismo valor al conseguido inicialmente sin obstáculos.

Esto es debido a cómo están definidos los pasos del robot. Si existe una cantidad inferior de pasos definidos en una dirección, éstos podrán tomarse a lo largo del recorrido de las demás, sin necesidad de efectuarse inmediatamente. Este hecho, da lugar a numerosas combinaciones de caminos con la misma distancia.

Por otro lado, el tiempo de ejecución se incrementa a medida que se incorporan más obstáculos. Esta es la causa por la que se requiere más tiempo para generar el espacio de trabajo, aunque la búsqueda de la ruta se realiza en menos tiempo al analizarse menos nodos.

## 5.2. Ejemplo 2. Aplicación industrial

En este ejemplo, se muestra una simulación en un entorno industrial. Desde el inicio, la configuración que presenta el robot se encuentra comprometida por la situación de los obstáculos. También se encuentra propuesto en (Abu-Dakka, et al. 2012).

**Tabla 20. Parámetros del ACO para el ejemplo 2.**

Hormigas	Iteraciones	$\alpha$	$\beta$	$\rho$	$\tau_0$
20	50	1	1	0,1	1

Como en el ejemplo anterior, se muestran las configuraciones inicial y final en la Tabla 22. Los obstáculos se definen en la Tabla 21.

**Tabla 21. Ubicación de los obstáculos.**

	1° Cilindro	2° Cilindro	3° Cilindro	4° Cilindro
Centro 1	(-0.70,0.50,0.00)	(-0.70,0.00,0.00)	(-0.70,-0.15,0.70)	(-0.70,0.15,0.15)
Centro 2	(-0.70,0.50,0.80)	(-0.70,0.00,0.80)	(-0.70,0.65,0.70)	(-0.70,0.65,0.15)
Radio	0.15	0.15	0.15	0.15
	1° Plano	2° Plano	3° Plano	4° Plano
Punto 1	(0.31,0.79,1.42)	(0.31,0.79,1.42)	(-0.03,0.79,1.42)	(-0.03,0.79,0.97)
Punto 2	(0.31,0.99,1.42)	(0.31,0.99,1.42)	(-0.03,0.99,1.42)	(-0.03,0.99,0.97)
Punto 3	(0.31,0.79,0.97)	(-0.03,0.99,1.42)	(-0.03,0.99,0.97)	(0.31,0.99,0.97)
Punto 4	(0.31,0.99,0.97)	(-0.03,0.79,1.42)	(-0.03,0.79,0.97)	(0.31,0.79,0.97)

**Tabla 22. Configuración inicial y final del robot.**

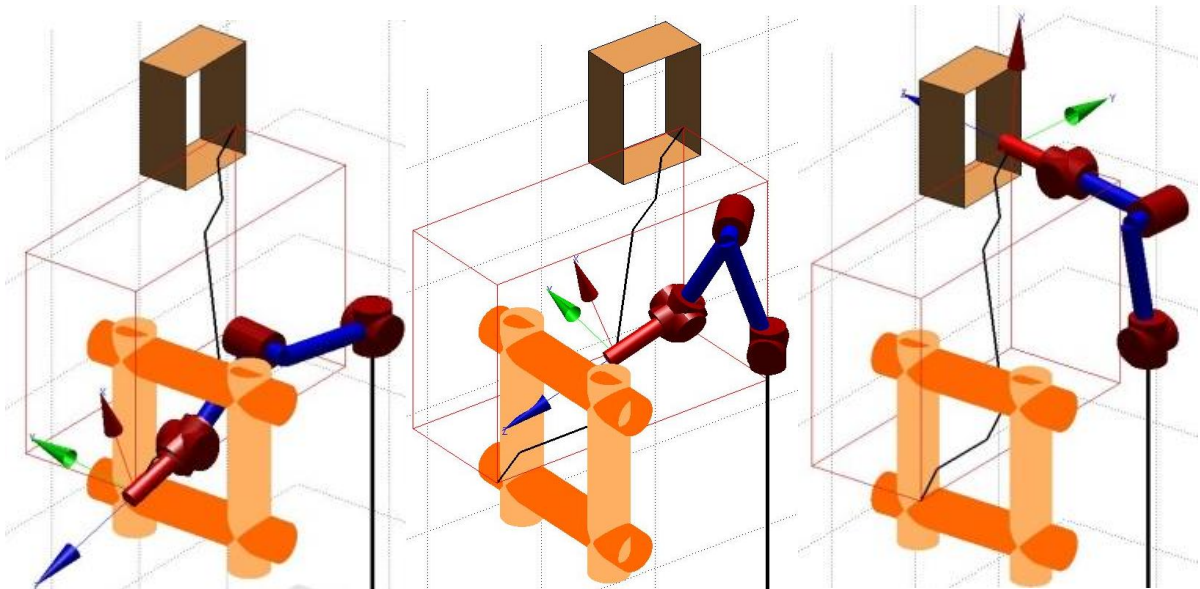
Articulación	1	2	3	4	5	6
$C^{inicial}(\mathbf{q})$	-7,50°	-174,80°	46,40°	4,30°	16,50°	-6,50°
$C^{final}(\mathbf{q})$	-95,10°	-101,20°	15,59°	0,00°	0,00°	0,00°

Los resultados se muestran en la Tabla 23. El rendimiento obtenido en este ejemplo es notablemente mejor al obtenido en el caso del uso de algoritmos genéticos en (Abu-Dakka, et al. 2012), donde la mejor ruta hallada tiene una longitud de 1.7175 [m].

**Tabla 23. Resultados ejemplo aplicación industrial, comparado con el artículo de (Abu-Dakka, et al. 2012), donde se usan algoritmos genéticos.**

ACO	Sin obstáculos	Con obstáculos
Distancia [m]	1.6548	1.6548
Tiempo ejecución [s]	6.7674	11.1096
Algoritmo genético	Sin obstáculos	Con obstáculos
Distancia [m]	1.8067	1.7175
Tiempo ejecución [s]	2156	11366

La Figura 32 muestra el progreso que sigue el robot por la ruta óptima conseguida.

**Figura 32. Movimiento del robot por la ruta establecida como óptima.**

### 5.3. Ejemplo 3. Obstrucción de rutas

Esta simulación estudia la respuesta que propone el algoritmo a medida que se posicionan nuevos obstáculos en sus rutas. De este modo, cada vez que se genera un camino, se sitúa un nuevo obstáculo para que, durante el siguiente análisis, se impida su recorrido.

Las configuraciones inicial y final se muestran en la Tabla 24. En la Tabla 25, se muestra la ubicación de los 16 obstáculos (esferas) que se han definido antes de que el algoritmo no pueda converger en una solución.

**Tabla 24. Configuración inicial y final del robot.**

Articulación	1	2	3	4	5	6
$C^{inicial}(\mathbf{q})$	-27.57	-28.26	148.61	0.50	-31.96	0.00
$C^{final}(\mathbf{q})$	131.71	-43.85	97.74	3.61	27.71	0.00

**Tabla 25. Ubicación de los obstáculos.**

Obstáculos	Centro	Radio
Esfera 1	(-0.0607905, 0.484387, 1.281231)	0,1
Esfera 2	(-0.0607905, 0.299922, 1.28123)	0,1
Esfera 3	(-0.0607905, 0.392155, 1.18158)	0,1
Esfera 4	(-0.0607905, 0.576619, 1.18158)	0,1
Esfera 5	(-0.0607905, 0.20769, 1.18158)	0,1
Esfera 6	(-0.0607905, 0.299922, 1.08192)	0,1
Esfera 7	(-0.0607905, 0.115458, 1.28123)	0,1
Esfera 8	(-0.0607905, 0.0232255, 1.18158)	0,1
Esfera 9	(-0.0607905, 0.484387, 1.08192)	0,1
Esfera 10	(-0.0607905, 0.115458, 1.08192)	0,1
Esfera 11	(-0.0607905, 0.392155, 0.982266)	0,1
Esfera 12	(-0.0607905, 0.576619, 0.982266)	0,1
Esfera 13	(-0.0607905, 0.484387, 0.88261)	0,1
Esfera 14	(-0.0607905, 0.576619, 0.782954)	0,1
Esfera 15	(-0.0607905, 0.392155, 0.782954)	0,1
Esfera 16	(-0.0607905, 0.484387, 0.683298)	0,1

Los parámetros definidos para el ACO:

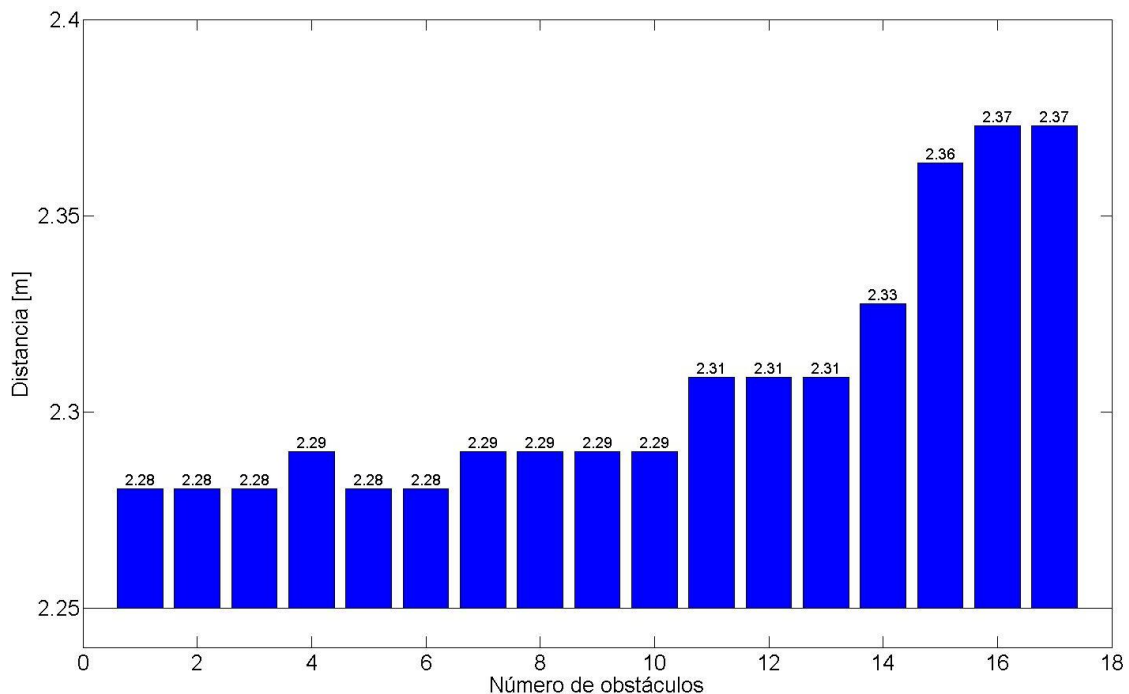
**Tabla 26 Parámetros del ACO para el ejemplo 3.**

Hormigas	Iteraciones	$\alpha$	$\beta$	$\rho$	$\tau_0$
20	1150	1	1	0,1	1

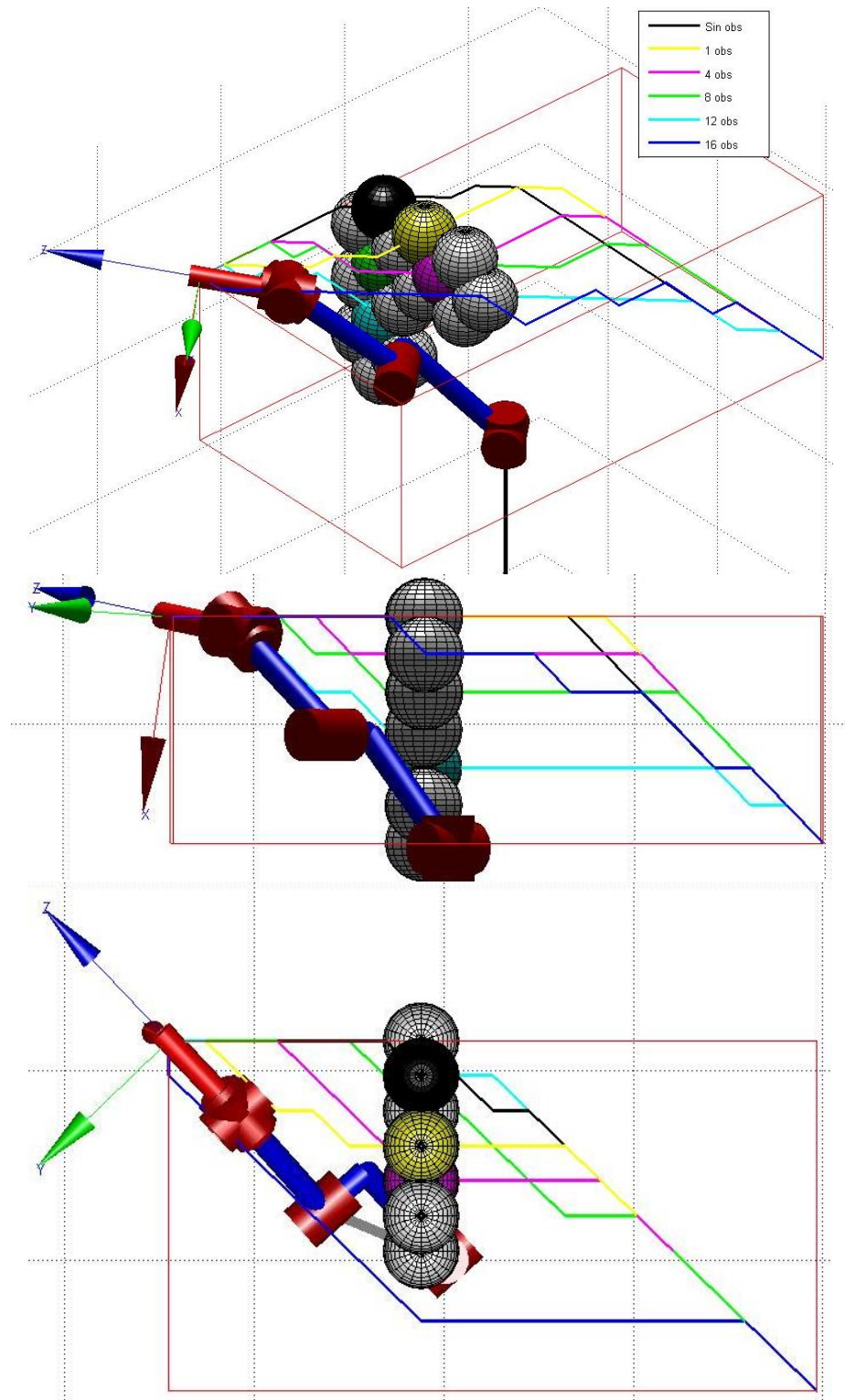
La visualización de los resultados se puede contemplar en la Figura 34 (se muestran 6 rutas de las 17 calculadas). Cuando el color de la ruta y el del obstáculo coinciden, implica que este último se ha definido a posteriori para forzar al algoritmo a buscar un nuevo camino.

En la Figura 34 se muestran las distancias recorridas en cada ruta como función de los obstáculos definidos en el espacio operacional. Cabe destacar que, con el posicionamiento de un tercer obstáculo, la ruta no debería aumentar su recorrido. Lo ha hecho porque ha caído en un mínimo local. Se está desarrollando una función que facilite una salida de este estancamiento.

Considerando que algoritmo desarrollado aún no es capaz de salirse de los límites del espacio de trabajo definido, sólo se admiten 16 obstáculos. No obstante, queda propuesto para futuros proyectos.



**Figura 33. Resultados del ejemplo de obstrucción de rutas, mostrando la distancia de cada una de las rutas en función de los obstáculos definidos.**



**Figura 34. Evolución de las rutas en el entorno. Se muestran todos los obstáculos con 5 rutas distintas para facilitar su visualización.**



# Capítulo 6

## Conclusiones

En este TFG se ha desarrollado un algoritmo, basado en el ACO, para el cálculo de rutas con manipuladores robóticos minimizando la distancia total que recorren. Todo ello, partiendo desde una configuración inicial del robot a una de destino y evitando colisiones con los obstáculos que se encuentran distribuidos por el espacio operacional. Como paso previo, y para optimizar el algoritmo, se ha realizado un estudio estadístico con el que se ha determinado la influencia que cada parámetro del ACO tiene en el tiempo de ejecución y en la evaluación de la ruta más corta (peso de los niveles de feromona y de su ratio de evaporación, peso de los niveles heurísticos, número de hormigas y número de iteraciones).

Los resultados obtenidos con el estudio estadístico permiten asegurar que la exploración y el aprendizaje deben tener una contribución similar, de modo que, el valor de  $\alpha$  y  $\beta$  debe ser semejante. El incremento en el número de hormigas permite una exploración más amplia, por lo que, en espacios muy grandes conviene potenciar este factor. El número de



iteraciones tiene un límite, a partir del cual, se comprueba que no hay una mejora en la optimización de las rutas porque el algoritmo converge en una solución única.

Se han desarrollado tres entornos distintos que ponen a prueba al ACO. En dos de ellos se han comparado los resultados obtenidos en este TFG con el estudio mencionado anteriormente, que hace uso de algoritmos genéticos. En estas aplicaciones concretas, se ha demostrado que el ACO, es capaz de encontrar rutas más cortas y todo ello en un menor tiempo de computación. Además, en un tercer caso, y yendo más lejos que en los estudios anteriores, se ha ampliado el espacio operacional a optimizar así como el número de obstáculos presentes (añadiendo obstáculos en cada una de las rutas que se iban obteniendo, que es la comprobación más extrema que podemos plantear). El ACO desarrollado en este TFG ha sido capaz de dar una solución eficiente a todos los problemas, ofreciendo en todos los casos la alternativa más lógica y eficaz.

Para **trabajos futuros**, se plantea:

- La realización de las variaciones del ACO (ACS, MMAS, etc.), para comprobar en qué casos pueden mejorar los resultados obtenidos, sin excluir que también podrían empeorarlos.
- Un método que permita salir de los mínimos locales que, aunque ha ocurrido puntualmente, podría repetirse por lo que habría que solucionar los posibles estancamientos del algoritmo.
- Hacer que las hormigas artificiales puedan generar nuevos nodos dentro del espacio operativo, para poder aplicar el método de generación parcial del espacio explicado en el capítulo 3.2.
- Conseguir un espacio de trabajo que pueda expandirse fuera de los límites definidos inicialmente por el prisma que se genera. De forma que si no existen rutas posibles, se pueda ampliar el espacio dando la posibilidad de encontrar otras que sí sean viables.

- Y por último, conseguir la metodología de entornos dinámicos, con el objetivo de que se pueda realizar la planificación de caminos de forma local.

# Capítulo 7

## Referencias

Abu-Dakka FJ, Valero F, Mata V. 2012. Evolutionary path planning algorithm for industrial robots. *Advanced Robotics*. 26(11-12):1369-1392.

Abu-Dakka FJ, Rubio F, Valero F, Mata V. 2013. Evolutionary indirect approach to solving trajectory planning problem for industrial robots operating in workspaces with obstacles. *European Journal of Mechanics-A/Solids*. 42:210-218.

Abu-Dakka FJM. 2011. Trajectory planning for industrial robot using genetic algorithms Valencia: Politécnica de Valencia.

Barraquand J, Latombe J-C. 1991. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*. 10(6):628-649.

Brooks RA. 1983. Planning collision-free motions for pick-and-place operations. *The International Journal of Robotics Research*. 2(4):19-44.

- Bullnheimer B, Hartl RF, Strauss C. 1997. A new rank based version of the Ant System - A computational study. In: Vienna: SFB Adaptive Information Systems and Modelling in Economics and Management Science, WU Vienna University of Economics and Business.
- Craig JJ. 2005. Introduction to robotics: mechanics and control. Pearson Prentice Hall Upper Saddle River.
- Chakraborty S. 2013. Ant colony system: A new concept to robot path planning. International Journal of Hybrid Information Technology. 6(6):11-30.
- Dahl ED. Neural network algorithm for an NP-complete problem: Map and graph coloring. Proceedings of the IEEE International Conference on Neural Networks; 1987: IEEE New York.
- Deng X, Zhang L, Luo L. 2013. An improved ant colony optimization applied in robot path planning problem. Journal of Computers. 8(3):585-593.
- Dorigo M. 1992. Optimization, learning and natural algorithms Italy: Politecnico di Milano.
- Dorigo M, Gambardella LM, Maria L. 1997. Ant colonies for the traveling salesman problem. Biosystems. 43(2):73-81.
- Dorigo M, Birattari M, Blum C, Clerc M, Stützle T, Winfield A. 2008. Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008, Brussels, Belgium, September 22-24, 2008, Proceedings. Springer.
- Garro BA, Sossa H, Vazquez RA. Evolving ant colony system for optimizing path planning in mobile robots. Proceedings of the Electronics, Robotics and Automotive Mechanics Conference; 2007; Morelos: IEEE.
- Goss S, Aron S, Deneubourg J-L, Pasteels JM. 1989. Self-organized shortcuts in the Argentine ant. Naturwissenschaften. 76(12):579-581.
- Guo Y, Shen X, Zhu Z. 2014. Research on the mobile robots intelligent path planning based on ant colony algorithm application in manufacturing logistics. International Journal of Computer Science & Information Technolo. 6(1):17.
- Hammerl T. 2009. Ant colony optimization for tree and hypertree decompositions Vienna: University of Technology.
- Hart PE, Nilsson NJ, Raphael B. 1968. A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics. 4(2):100-107.

Hwang YK, Ahuja N. 1992. Gross motion planning—a survey. *ACM Computing Surveys (CSUR)*. 24(3):219-291.

Kavraki LE, Svestka P, Latombe JC, Overmars MH. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*. 12(4):566-580. doi 10.1109/70.508439

Khatib O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*. 5(1):90-98.

Klee V. 1980. On the complexity of  $d$ -dimensional Voronoi diagrams. *Archiv der Mathematik*. 34(1):75-80.

LaValle SM. 1998. Rapidly-exploring random trees: A new tool for path planning. Iowa State University.

Li Z, Chen X, Xiao W. 2004. A new motion planning approach based on artificial potential field in unknown environment. Berlin Heidelberg: Springer.

Lozano-Pérez T, Wesley MA. 1979. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*. 22(10):560-570. doi 10.1145/359156.359164

Mei H, Tian Y, Zu L. 2006. A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment. *International Journal of Information Technology*. 12(3):78-88.

Mohammed KS. 2013. Modified Ant Colony Optimization for Solving Traveling Salesman Problem. *International Journal of Engineering & Computer Science IJECS-IJENS*. 13(5):1-5.

Negulescu SC, Oprean C, Kifor CV, Carabulea I. Elitist ant system for route allocation problem. *Proceedings of the Proceedings of the 8th Conference on Applied Informatics and Communications*; 2008; Rhodes, Greece.

Osher S, Sethian JA. 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*. 79(1):12-49.

Sariff NB, Buniyamin N. Ant colony system for robot path planning in global static environment. *Proceedings of the 9th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE'10)*; 2010.

Stützle T, Hoos HH. 2000. MAX-MIN ant system. *Future generation computer systems*. 16(8):889-914.

Tewolde GS, Sheng W. 2008. Robot Path Integration in Manufacturing Processes: Genetic Algorithm Versus Ant Colony Optimization. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*. 38(2):278-287. doi 10.1109/TSMCA.2007.914769

Warren CW. Fast path planning using modified A\* method. *Proceedings of the Robotics and Automation, 1993 Proceedings, 1993 IEEE International Conference on*; 1993; Atlanta, GA: IEEE.

Zhang X, Zhong Q. Research on Path Planning for Humanoid Robot based on Ant Colony Algorithm. *Proceedings of the 3rd International Conference on Computer Science and Service System*; 2014; Bangkok, Thailand: Atlantis Press.

Zhao M, Yong D. 2015. Robot Three Dimensional Space Path-planning Applying the Improved Ant Colony Optimization. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 14(2):304-310.

Zhou D, Wang L, Zhang Q. 2016. Obstacle avoidance planning of space manipulator end-effector based on improved ant colony algorithm. *SpringerPlus*. 5(1):1-13. doi 10.1186/s40064-016-2157-x

Zhou S, Li Y, Li X. 2014. An Improved Ant Colony Optimization for the Multi-Robot Path Planning with Timeliness. *International Journal of Smart Home*. 8(2):201-210.

# Apéndice A

## Presupuesto

En este anexo se presenta toda la planificación llevada a cabo para el desarrollo de este TFG, así como su evaluación económica.

- Coste de licencias y materiales:
  - Precio estimado del ordenador donde se ha realizado todo el trabajo 600€.
  - Precio de compilador para desarrollar el algoritmo en C++, 0€ dado que se trata de un software libre.
  - Precio de licencia de uso académico de Matlab 500€.
- Un salario por hora de 7€ suponiendo que me encontrase ejerciendo de titulado de grado medio.
- Precio valorado por hora de tutor 25€.

**Tabla 27. Horas estimadas de trabajo en el proyecto.**

Descripción de la tarea	Horas (estimadas)
Investigación inicial sobre el Path Planning y el ACO	20
Presentación y exposición del funcionamiento del ACO, así como de sus aplicaciones	20
Comparación de distintos códigos y diagramas de clase para algoritmos basados en el ACO	10
Entendimiento de la librería escogida	30
Creación de algoritmo para discretización de espacios operativos	40
Modificación de la librería para acoplar el nuevo espacio operacional	50
Simulación de distintos entornos	20
Estudio estadístico de los parámetros	40
Profundizar en cálculos de cinemática de robot	10
Implementación de librerías cinemáticas	25
Implementación de librerías de obstáculos	20
Simulación y representación de los resultados	30
Generación de imágenes	10
Redacción de la memoria	160
Escritura de un artículo para la Revista Iberoamericana de Automática e Informática industrial	25
<b>Total</b>	<b>510</b>

**Tabla 28. Coste del Proyecto.**

Descripción	€/hora	Horas	Importe (€)
Tiempo empleado por el autor	7	510	3570
Tiempo empleado por el tutor	25	175	4375
Coste material y de licencias	-	-	1100
<b>Total</b>			<b>9045</b>